

**FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO**



# **Projeto de Malha de Captura de Fase Autocalibrável**

**Rui André Moutinho Teixeira**

**DESIGN OF A HIGH PERFORMANCE N-INTEGER SELF-TESTABLE AND  
SELF-CALIBRATABLE PHASE-LOCKED LOOP**

**MESTRADO INTEGRADO EM ENGENHARIA ELECTROTÉCNICA E DE  
COMPUTADORES**

**Orientador: Professor José Machado da Silva**

**2 de Agosto de 2018**



# Resumo

A significativa evolução das tecnologias da microeletrônica e os cada vez maiores níveis de integração de semicondutores por unidade de área, permitem a integração de circuitos mistos (digitais e analógicos) de elevado desempenho — menores consumos relativos e maiores frequências de operação. Porém, estes circuitos são também mais sensíveis à variabilidade dos parâmetros funcionais (devido a variações do processo de fabrico e a degradação com o tempo de funcionamento) e a efeitos de grandezas externas, particularmente temperatura e tensão de alimentação — o conjunto dos desvios usualmente designados por PVT (*Process, Voltage, Temperature*).

Os circuitos digitais tendem a apresentar maior tolerância a erros devidos a estes desvios, contudo em circuitos analógicos e de radiofrequência podem ocorrer falhas funcionais notórias para as mesmas condições PVT.

O processo industrial que permite testar todas estas condições é de alto rigor, moroso, e difícil, elevando assim o seu custo final de fabrico. Uma forma de mitigar este problema consiste em recorrer a circuitos de autoteste e de autocalibração — também conhecidos por Built-in Self-Test (BIST) e Built-In Self-Calibration. Como já é conhecido, este tipo de circuitos está bem estabelecido no domínio dos circuitos digitais e dada esta necessidade têm vindo a ser usados também em circuitos analógicos e mistos.

Esta dissertação tem como objetivo desenvolver uma metodologia de autocalibração para um circuito Phase-Locked Loop (PLL) aplicando algoritmos de *Machine Learning* do tipo *feature selection* e *pattern recognition*, mais concretamente *backward-forward selection* e processo de regressão gaussiana, para correção de parâmetros fundamentais do funcionamento do circuito. Recorre-se à programação e integração de um modelo matemático criado em Matlab, com o intuito de estabelecer um conhecimento "a priori" dos fatores relevantes a desvios PVT que afetam uma função objetivo, em sequência do qual é estabelecido um processo que permite autotestar e autocalibrar a PLL. O circuito que realiza este processo foi implementado e validado com um modelo VHDL, dando resultado a um módulo RTL *register-transfer level* sintetizável.



# Abstract

Given the significant evolution of microelectronics technology over the years, as it was previously foreseen by the famous Moore's Law, semiconductors have become increasingly smaller in size allowing integration of digital, analogue and mixed circuits with better performance, such as lower power consumption and higher operating frequency. However, it is not always possible to maintain the conditions required to guarantee the good performance level necessary for its proper functioning.

The deviation from the desired operation is generally due to adverse conditions generated by factors related to the fabrication process, as well as temperature and power supply voltage variations — the so called PVT (Process, Voltage, Temperature) variations. The digital circuits tend to have a greater margin of acceptance to these deviations, but in the analogue and radio frequency circuits domain, critical failures may occur for the same PVT conditions leading to wear and malfunction.

The industrial process that allows to test all these conditions is time consuming, difficult to implement and of high rigor, thus increasing the cost involved in maintaining this type of circuits in optimal operating conditions. The solution to this problem has resorted to the construction of self-test and self-calibration circuits, also known as Built-in Self-Test (BIST) and Built-In Self-Calibration (BISC). As already known, these types of circuits are well established in the field of digital circuits and given the above identified need, there has been an increasing request to adopt them also in the analogue and mixed-signal domains.

This dissertation aims to develop and implement self-calibration mechanisms for a Phase-locked-loop (PLL) circuit by applying Machine Learning feature selection and pattern recognition algorithms, namely backward-forward selection and gaussian regression processes, to characterize fundamental parameters of the proposed circuit. Using the programming and integration of a mathematical model created in MatLab, in order to establish an "a priori" knowledge of the relevant PVT deviations factors that affect an objective function, a system is then established that allows to self-test and self-calibrate the PLL. The circuit that realizes this process was implemented and validated with a VHDL model, resulting in a full synthesible register transfer level module.



# Agradecimentos

Gostaria de agradecer ao meu orientador, professor José Machado da Silva, por toda a orientação e apoio neste trabalho. Além disso, agradeço-lhe pela sua prestabilidade durante todo o meu percurso académico.

Agradeço aos colegas do laboratório por toda ajuda técnica, sendo que a sua opinião foi crucial para várias decisões tomadas no trabalho.

Agradeço à minha família, especialmente aos meus pais e à minha irmã, por todo o apoio moral e por serem sempre disponíveis para me ajudar com tudo.

Agradeço aos meus amigos e colegas pelo incentivo e pelo interesse demonstrado por este trabalho.

Agradeço especialmente à Solange pelo apoio emocional e por estar sempre do meu lado.

Rui Teixeira





*“I don’t care that they stole my idea . . .  
I care that they don’t have any of their own.”*

Nikola Tesla, electrical engineer and physicist.



# Conteúdo

<b>1</b>	<b>Introdução</b>	<b>1</b>
1.1	Contextualização . . . . .	1
1.2	Motivação . . . . .	2
1.3	Objetivos . . . . .	2
1.4	Estrutura do Documento . . . . .	3
<b>2</b>	<b>Metodologias de Auto Teste e Calibração — uma revisão</b>	<b>5</b>
2.1	Circuitos Analógicos e mistos . . . . .	5
2.1.1	Efeitos PVT numa <i>PLL</i> . . . . .	9
2.2	Métodos de teste - Aplicações e BIST . . . . .	11
2.2.1	Circuitos auxiliares de teste . . . . .	11
2.2.2	Redes de sensores e <i>DSP</i> . . . . .	12
2.2.3	Aplicação de <i>PSO</i> para reconfiguração automática de <i>Hardware</i> . . . . .	13
2.2.4	Cancelamento de efeitos PVT usando sistemas de autoteste . . . . .	15
2.2.5	Considerações de autocalibração para sintetizadores de frequência . . . . .	17
2.3	Algoritmos de seleção de variáveis e reconhecimento de padrões . . . . .	20
2.3.1	<i>Feature selection</i> . . . . .	21
2.3.2	Redução do erro de frequência em MEMS . . . . .	23
<b>3</b>	<b>Fundamentos de desenho da PLL</b>	<b>27</b>
3.1	Modelo linear . . . . .	29
3.1.1	Projeto da PLL a usar como caso de estudo . . . . .	30
3.1.2	Requisitos do sistema . . . . .	30
3.1.2.1	Detetor de fase . . . . .	30
3.1.2.2	Charge-pump . . . . .	31
3.1.2.3	Filtro . . . . .	32
3.1.2.4	Oscilador controlado por tensão . . . . .	33
3.1.2.5	Divisor . . . . .	34
3.1.3	Análise dinâmica . . . . .	34
3.1.4	Resultados simulação MATLAB . . . . .	36
3.1.5	Resultados de simulação em CppSim . . . . .	41
3.2	Implementação dos blocos . . . . .	46
3.2.1	Detetor de fase . . . . .	46
3.2.2	Charge-pump de duplo andar . . . . .	47
3.2.3	Filtro <i>lead-lag</i> . . . . .	50
3.2.4	Oscilador em anel . . . . .	51
3.2.5	Topologia do divisor-N inteiro . . . . .	55
3.2.6	Resultados da implementação final . . . . .	58

<b>4</b>	<b>Calibração e Testabilidade</b>	<b>63</b>
4.1	Metodologia . . . . .	64
4.1.1	Análise <i>a priori</i> . . . . .	65
4.1.2	Análise <i>a posteriori</i> . . . . .	69
4.1.3	Módulos lógico Verilog HDL . . . . .	73
4.1.3.1	Módulo do oscilador controlado por tensão . . . . .	73
4.1.3.2	Módulo do controlador da calibração . . . . .	74
4.1.3.3	Síntese módulo controlador . . . . .	75
4.1.3.4	Simulação do algoritmo em Verilog HDL . . . . .	76
4.2	Avaliação da cobertura da calibração . . . . .	78
<b>5</b>	<b>Conclusões e Trabalho Futuro</b>	<b>83</b>
5.1	Trabalho Futuro . . . . .	84
<b>A</b>	<b>Resultados implementação com ganho do oscilador inferior ao da implementação final</b>	<b>85</b>
<b>B</b>	<b>Resultados simulação Verilog HDL</b>	<b>89</b>

# Lista de Figuras

2.1	Sinal analógico em (a) e sinal digital em (b). . . . .	5
2.2	Ruído em um ADC [1]. . . . .	6
2.3	PSRR vs ruído da fonte em frequência para o ADC12040 [1]. . . . .	6
2.4	Alteração da dependência da temperatura de $V_{TH}$ pela espessura do óxido $d$ e dopagem $N_A$ [2]. . . . .	7
2.5	Dependência da corrente, devido a variações PVT numa tecnologia de 90 nm [2].	7
2.6	Contribuições na variabilidade do fabrico de CMOS em escala nanométrica [3]. .	8
2.7	Classificação das variações de processo [4]. . . . .	9
2.8	Alterações no desempenho da PLL por efeitos PVT [3]. . . . .	10
2.9	Impacto de variações VT na frequência à saída do VCO [5]. . . . .	10
2.10	Impacto de variações PVT na banda de passagem em malha fechada de uma PLL [6].	11
2.11	Exemplo de diagrama de blocos de um mecanismo de <i>BIST</i> [7] . . . . .	12
2.12	<i>Sensor network</i> usando a um DSP para calibração de um circuito RF [3] . . . . .	12
2.13	Metodologia para conceção em sistemas inteligentes de baixa potência reconfiguráveis [8]. . . . .	13
2.14	Algoritmo usado em <i>PSO</i> [8]. . . . .	14
2.15	Diagrama de blocos de um modelo de classificação 1-NN [8]. . . . .	15
2.16	Dependência do <i>jitter</i> em função da corrente média de comutação e da capacidade de desacoplamento da fonte de alimentação [9]. . . . .	16
2.17	Circuito proposto para compensação e deteção PVT [9]. . . . .	16
2.18	Resultados do mecanismo de autoteste apresentado em [10]. . . . .	17
2.19	Metodologia principal proposta em [11]. . . . .	18
2.20	Comparação do VCO relativamente a uma frequência de referência. (a) Circuito proposto. (b) Princípio de operação [11]. . . . .	18
2.21	Convergência do princípio de autocalibração por meio de comutação de capacidades [11]. . . . .	19
2.22	Autocalibração de um oscilador LC [12]. . . . .	20
2.23	Comparação gráfica do <i>feature set</i> , sendo a região mais estreita a representação da evolução dos algoritmos [13]. . . . .	22
2.24	Desenho de um oscilador MEMS [14] . . . . .	23
2.25	Redução do erro da frequência aplicando algoritmos de <i>Sequential backward-forward selection</i> [14]. . . . .	24
3.1	Arquitetura de uma PLL [15]. . . . .	27
3.2	Diagrama dos passos de desenho da PLL. . . . .	28
3.3	Relação do <i>Lock Range</i> e <i>Capture Range</i> em torno da frequência de oscilação. . .	29
3.4	Exemplo de detetor de fase de 3 estados [16]. . . . .	30
3.5	Exemplo de funcionamento do detetor de fase 3.4. . . . .	31

3.6	Funcionamento da <i>charge-pump</i> . . . . .	31
3.7	Filtro passa baixo <i>lead-lag</i> [17]. . . . .	32
3.8	Varrimentos em função de $N_{clocks}$ . . . . .	37
3.9	Simulações iniciais considerando $H(s)$ um sistema de 3ª ordem. . . . .	38
3.10	Varrimento para $\omega_n$ superior. . . . .	39
3.11	Aproximação de $H(s)$ a um sistema de 2ª ordem. . . . .	39
3.12	$H(s)$ sem considerar aproximação de 2ª ordem. . . . .	40
3.13	$H(s)$ implementado. . . . .	40
3.14	Processo de modelação dos blocos em CppSim® [15]. . . . .	41
3.15	Esquemático do modelo CppSim da PLL[15]. . . . .	42
3.16	Resposta a um degrau de 1 Hz em $F_{ref}$ para os resultados em 3.35. . . . .	43
3.17	Resposta a um degrau de 1 Hz em $F_{ref}$ — resultados em 3.32 com a exceção de $K_{vco} = 17$ kHz. . . . .	43
3.18	Resposta ao degrau para $K_{vco} = 17$ kHz. . . . .	44
3.19	Resposta ao degrau para $K_{vco} = 65$ kHz para a parametrização final em 3.32. . . . .	44
3.20	Resposta a um degrau de 1 Hz em $F_{ref}$ para os valores de 3.36 . . . . .	45
3.21	Detetor de fase [18]. . . . .	46
3.22	Circuito do <i>charge-pump</i> [19]. . . . .	47
3.23	Funcionamento <i>charge-pump</i> [19]. . . . .	48
3.24	Amplificador operacional. . . . .	49
3.25	Caracterização da corrente $I_{cp}$ do <i>charge-pump</i> . . . . .	50
3.26	Diagrama de Bode para o filtro. . . . .	50
3.27	Caracterização da resultante do ganho $K_{pd}$ e do ganho do filtro. . . . .	51
3.28	Circuito da célula de atraso e a respetiva fonte de corrente. . . . .	52
3.29	Corrente aos terminais do andar <i>current starved</i> . . . . .	53
3.30	Esquemático VCO [5]. . . . .	54
3.31	$K_{vco}$ do oscilador. . . . .	54
3.32	Circuito da malha N de divisores. . . . .	55
3.33	Formas de onda nas diversas saídas do divisor N. . . . .	56
3.34	Circuito da seleção das frequências à saída. . . . .	56
3.35	Circuitos lógicos do bloco correspondente à divisão de frequência [20]. . . . .	57
3.36	Circuito divisor $\div 3$ [20]. . . . .	57
3.37	<i>Jitter</i> de frequência do sinal ( $f_{div}$ ) de 50 mHz à entrada do detetor de fase. . . . .	58
3.38	Incerteza do instante de ocorrência da transição à frequência de 50 Hz. . . . .	59
3.39	Incerteza do instante de ocorrência da transição à frequência de 600 Hz. . . . .	59
3.40	Resposta transitória do oscilador ao degrau. . . . .	60
3.41	Resposta ao degrau do sistema implementado 3.32 à saída do VCO. . . . .	61
3.42	Resposta ao degrau do sistema implementado 3.32 visto na frequência de 50 Hz. . . . .	61
3.43	Consumo corrente visto da fonte tensão $V_{dd}$ . . . . .	62
3.44	Consumo corrente visto da fonte tensão $V_{dd}$ (ampliada). . . . .	62
4.1	Diagrama de blocos generalizado da implementação proposta. . . . .	65
4.2	Variações de processo para 1015 pontos de dados. . . . .	67
4.3	Impacto das variações de temperatura para o intervalo entre $-40^\circ\text{C}$ e $125^\circ\text{C}$ . . . . .	68
4.4	Variações de PVT para 1015 pontos de dados. . . . .	68
4.5	Célula de atraso programável proposta para calibração do VCO. . . . .	70
4.6	Fluxograma do algoritmo de calibração. . . . .	72
4.7	Diagrama de blocos Verilog HDL de verificação do processo de calibração. . . . .	73
4.8	Linearização $K_{vco}$ em relação ao 4.4. . . . .	74

4.9	Passo de calibração para $K_{vco}$ $1,5\times$ superior, modo=5. . . . .	77
4.10	Passo de calibração para $K_{vco}$ $2\times$ inferior, modo=0. . . . .	77
4.11	Melhoria do <i>jitter</i> observado à frequência de 50 Hz, devido a efeitos PVT. . . . .	79
4.13	Melhoramento do <i>jitter</i> com a ativação de mais um divisor na cadeia e a reprogramação do VCO. . . . .	79
4.12	Calibração do ganho $K_{vco}$ após ajuste de $I_{dc}$ e $C_{apc}$ . . . . .	80
4.14	Melhoramento do desempenho da PLL, decorrente da calibração do $K_{vco}$ e comutação de $N_{div}$ . . . . .	81
A.1	$K_{vco} = 21.67$ KHz/V. . . . .	85
A.2	Resposta ao degrau. . . . .	86
A.3	Incerteza do instante de ocorrência da transição para o mesmo tempo de assentamento do $K_{vco} = 65$ KHz/V à frequência de 50 Hz. . . . .	86
A.4	Incerteza do instante de ocorrência da transição em <i>steady-state</i> à frequência de 50 Hz . . . . .	87
A.5	<i>Jitter</i> de frequência em <i>steady-state</i> . . . . .	87
A.6	Variação da frequência no oscilador durante a resposta transitória. . . . .	88
A.7	Banda máxima do sinal à entrada, com o exemplo de sinal com degrau de 9Hz para a implementação com maior banda da PLL. . . . .	88
B.1	Primeira iteração do algoritmo correspondente à calibração de um $K_{vco}$ 1.5X superior	89
B.2	Simulação completa do algoritmo correspondente à calibração de um $K_{vco}$ 1.5X superior. . . . .	90





# Lista de Tabelas

3.1	Resultado consoante o valor de $\Delta\phi$ para efeitos de carga no filtro. . . . .	32
3.2	Primeiras parametrizações do filtro em relação à figura 3.9. . . . .	38
3.3	Resultado em $Lag_{comp}$ pelas entradas $f_{ref}$ e $f_{div}$ . . . . .	46
3.4	Dimensionamento do detetor de fase. . . . .	46
3.5	Dimensionamento das portas lógicas da PLL. . . . .	47
3.6	Dimensionamento do <i>charge-pump</i> . . . . .	48
3.7	Dimensionamento do amplificador operacional. . . . .	49
3.8	Dimensões da célula de atraso. . . . .	52
3.9	Frequências à saída do bloco de divisor N. . . . .	55
4.1	Resultado sumário da síntese em valores estimados. . . . .	75
4.2	Resultado detalhado da síntese. . . . .	76
4.3	Contagem do nº de ciclos de relógio em relação ao relógio de referência $F_{ref}$ . . .	77
4.4	Contagem do N° de ciclos de relógio em relação ao relógio de referência $F_{ref}$ . . .	77



# Abreviaturas e Símbolos

## Lista de abreviaturas

ADC	Analog to digital converter
ASIC	Application Specific Integrated Circuit
BIST	Built-in self-test
CP	Charge-pump
CMP	Chemical Mechanical Planarization
CMOS	Complementary Metal Oxide Semiconductor
DSP	Digital Signal Processor
FDC	Frequency-digital conversion
FPGA	Field-Programmable Gate Array
LPF	Low-pass filter
LUT	Look up table
MEMS	Microelectromechanical System oscillator
MOSFET	Metal-oxide-semiconductor field-effect transistor
NAND	Not AND
NMOS	N-type metal-oxide-semiconductor logic
PFD	Phase-frequency detector
PLL	Phase-locked loop
PMOS	P-type metal-oxide-semiconductor logic
PSO	Particle swarm optimization
PSRR	Power supply rejection ratio
PVT	Process, Voltage and Temperature
RF	Radio-Frequency
RTL	Register-transfer level
SNR	Signal-to-noise ratio
SVM	Support Vector Machine
VCO	Voltage-controlled Oscillator
VHDL	VHSIC Hardware Description Language
VHSIC	Very High Speed Integrated Circuits
VLSI	Very-large-scale integration
XOR	Exclusive OR

## Lista de símbolos

$V_{TH}$	Threshold Voltage
$V_{GS}$	Gate-Source Voltage
$V_{DS}$	Drain-Source Voltage
$W$	Gate Width
$L$	Gate Length
$I_D$	Drain current
$C_{ox}$	Gate oxide capacitance per unit area
$\mu_n$	Charge-carrier effective mobility
$\lambda$	channel-length modulation parameter
$\phi_F$	Fermi energy, half the contact potential
$\phi_{gs}$	gate-substrate contact potential
$V_{FB}$	Flat band voltage
$\gamma$	Body effect parameter
$T$	Temperature
$k$	Boltzmann's constant
$q$	Elementary charge
$N_A$	Doping parameter
$N_i$	Intrinsic doping parameter for the substrate
$\omega_n$	Natural oscillation frequency
$\omega_z$	Transfer function zero frequency
$\omega_p$	Transfer function pole frequency
$\omega_c$	Crossover frequency

# Capítulo 1

## Introdução

Neste capítulo são abordados os aspetos introdutórios do trabalho a desenvolver na dissertação. É feita uma contextualização do problema e do âmbito em que se insere, é descrita a abordagem seguida na realização do projeto, e são referidos os motivos pelos quais foi escolhida o tema desta dissertação. Finalmente, são apresentados os objetivos do projeto e é descrita a estrutura do presente documento.

### 1.1 Contextualização

O uso de mecanismos de autoteste, também conhecidos por *built-in self-test* ou mais simplesmente *BIST*, é cada vez mais recorrente na produção de circuitos eletrónicos de alta fiabilidade, independentemente de o circuito ser de variável discreta ou contínua, isto é analógico, digital, ou misto. De um modo geral, estes mecanismos recorrem a cópias do circuito principal (funcionamento paralelo), a dados guardados em memória com informação previamente obtida ou capturada em tempo-real, ou a circuitos auxiliares de leitura e correção de dados.

O circuito a usar como caso de estudo neste trabalho é uma PLL (*Phase-locked-loop*). Este tipo de circuito é muito utilizado em sistemas de telecomunicações, rádio e computadores. São geralmente responsáveis pela geração de sinais de relógio, circuitos moduladores e desmoduladores, assim como em circuitos de recuperação de relógio em sistemas de comunicação série. A PLL permite a geração de um sinal de saída cuja fase se mantém em sintonia com a do sinal de entrada. A manutenção desta relação implica que a frequência do sinal de saída seja capaz de seguir a frequência do sinal de entrada mantendo-as iguais, isto é sincronizando-as. Além desta funcionalidade, o circuito permite gerar múltiplos da frequência de entrada funcionando assim como multiplicador ou divisor de frequência.

A verificação do funcionamento da PLL é daí um desafio nas áreas de testabilidade, nomeadamente no que diz respeito a encontrar soluções que sejam capazes de corrigir desvios que afetam o seu desempenho. Estes desvios estão geralmente associados a efeitos de naturezas diferentes sendo agrupados em variações de processo, temperatura e tensão. O modo como se manifestam numa PLL é também diferente, apresentado erros que afetam a qualidade do sinal, através de

ruído de fase e/ou espúrios, assim como a estabilidade do sistema, isto é, a sintonia e o tempo de estabelecimento após a resposta transitória.

O circuito analógico de uma PLL é responsável pela maior parte dos erros atrás referidos, pelo que é sobre estes que deve incidir o maior esforço de projeto no que diz respeito a minimizar a sua sensibilidade a efeitos PVT, e devem ser estes os primeiros visados nas metodologias de teste e calibração.

## 1.2 Motivação

Dada a necessidade de manter o desempenho de circuitos analógicos e mistos, aqui concretamente de PLL, dentro dos limites das tolerâncias estabelecidos, têm sido desenvolvidos métodos que permitem testar e calibrar estes sistemas. O estado da arte atual aponta para mecanismos que visam o uso de circuitos de autoteste (ou *BIST*), entre os quais se inclui os que realizam a observação e cálculo em tempo-real das variáveis PVT e do seu efeito no circuito. Esta abordagem requer a inclusão em grande escala de circuitos auxiliares fiáveis, dado que também estes podem apresentar defeitos, o que aumenta os custos de projeto e produção.

Nesta dissertação é proposto um método de autoteste e calibração de uma PLL, em que se procura minimizar as desvantagens do uso de circuitos *BIST* referidos no estado da arte, aplicando algoritmos de *Machine Learning* conhecidos, por meio da otimização dos modelos dos efeitos PVT e dos circuitos de correção a implementar, assim como minimizar o recurso a cálculo em tempo-real. Deste modo, os algoritmos de correção podem ser guardados num circuito de memória simples e é reduzida a complexidade dos circuitos auxiliares necessários.

## 1.3 Objetivos

O trabalho desenvolvido nesta dissertação foca-se essencialmente nos seguintes aspetos:

- Revisão dos mecanismos associados à ocorrência de defeitos e desvios das características de desempenho dos circuitos analógicos e mistos;
- Revisão da teoria do funcionamento e das características de desempenho das PLL, e levantamento do estado da arte de metodologias de teste e de projeto para a testabilidade de PLL;
- Aplicar algoritmos de *Machine learning* do tipo *feature selection*, mais concretamente *backward-forward selection*, para caracterização de parâmetros fundamentais do funcionamento do circuito proposto;
- Extrapolar condições de calibração e teste através do reconhecimento de padrões com recurso a um processo de regressão gaussiana;
- Criação e validação de um modelo matemático/lógico de autocalibração de uma PLL;
- Implementação e verificação do algoritmo em VHDL.

## 1.4 Estrutura do Documento

Este documento está estruturado em cinco capítulos, sendo o presente o de Introdução. Nos capítulos seguintes faz-se uma revisão bibliográfica, são apresentados os fundamentos de desenho de PLL usada como caso de estudo, descreve-se a estratégia de projeto para a calibração e a testabilidade proposta, e por fim apresentam-se as conclusões principais.

No presente capítulo, são abordados o enquadramento, a motivação e os objetivos da dissertação. Como revisão do Estado da Arte são apresentadas técnicas de caracterização descritas em artigos científicos publicados na última década.

No capítulo Fundamentos de Desenho da PLL será apresentado a metodologia para o projeto do desenho para o circuito em análise da PLL e a sua implementação em CMOS.

Em seguida é feita um estudo das variações PVT no circuito da PLL, tomando uma metodologia a dois passos, sendo inicialmente feita uma formulação *a priori*, recorrendo a simulação estatística para obtenção de um grande conjunto de dados de treino, e procede-se ao seu enquadramento por forma a criar padrões de treino e um modelo matemático que caracteriza com maior certeza o funcionamento mais provável. Com base neste, é concebido *a posteriori* um algoritmo que segue uma estrutura lógica por forma a calibrar parâmetros críticos do circuito, através do uso de princípios de sistemas reconfiguráveis. É por fim implementado o algoritmo numa descrição VHDL de modo a simular e obter resultados que demonstram a convergência para uma solução de autocalibração.

No último capítulo são apresentadas as conclusões finais e são delineadas propostas de trabalho futuro.





## Capítulo 2

# Metodologias de Auto Teste e Calibração — uma revisão

### 2.1 Circuitos Analógicos e mistos

Um circuito analógico ou misto é considerado um sistema de variável contínua, i. e., que expressa uma variação de uma grandeza, por exemplo corrente ou tensão, segundo uma função contínua no tempo. Num sinal analógico não existem descontinuidades de valores e não ocorrem alterações bruscas de um valor em relação ao anterior, contrastando com os níveis bem definidos e assumidos como instantâneos em termos lógicos associados aos circuitos digitais.

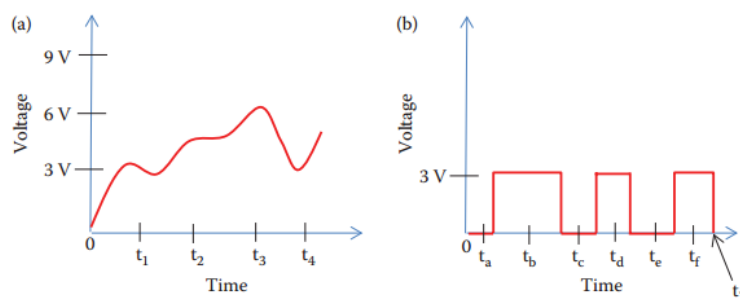


Figura 2.1: Sinal analógico em (a) e sinal digital em (b).

Sendo os valores manipulados por circuitos digitais representados por dois níveis distintos, entre os quais existe uma margem de separação, estes circuitos são mais robustos a variações das características dos seus componentes. Nos analógicos, o ponto de funcionamento contínuo acompanha e é afetado pelas não idealidades dos componentes, nomeadamente as devidas a efeitos PVT, tais como variações da tensão de *threshold* e consequentemente do ponto de funcionamento estático de transístores MOSFET, com consequências na diminuição ou aumento da corrente de polarização dos circuitos.

A título de exemplo, considere-se o circuito do andar de saída de um ADC (conversor analógico/digital) e o efeito do ruído de duas fontes de tensão.

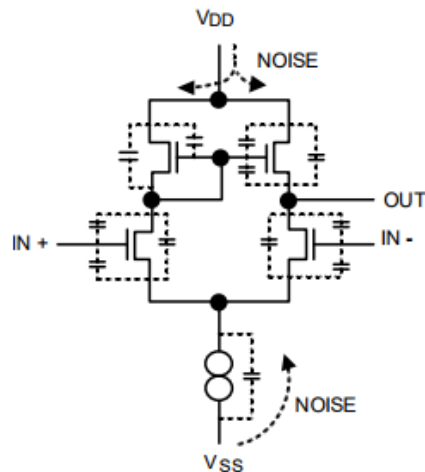


Figura 2.2: Ruído em um ADC [1].

O efeito transitório da alteração de tensões de alimentação ilustrado na figura 2.2 causa flutuações de tensão e corrente, provocando ruído de alta frequência que terá consequências na relação sinal ruído (SNR) do ADC.

É possível verificar na figura 2.3 que caso ocorram variações transitórias na tensão, a capacidade de rejeitar estas variações é menor com o aumento da frequência, isto é, o circuito é mais propício a ter degradação do seu funcionamento com variações de tensão em frequências mais altas.

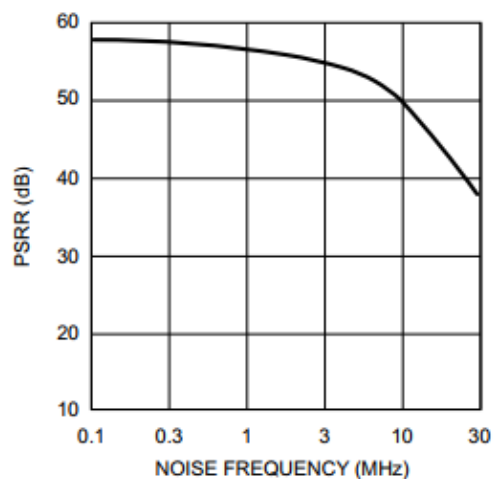


Figura 2.3: PSRR vs ruído da fonte em frequência para o ADC12040 [1].

Como referido, para além da tensão e da corrente é também de denotar o efeito da temperatura nos circuitos analógicos em geral, e nos MOSFET em particular. Revendo alguns conceitos acerca de semicondutores temos que, a corrente de polarização num MOSFET determina a sua região de

operação, podendo operar em modo de tríodo (a equação 2.1) ou saturação (equação 2.2), não considerando para este assunto a região de operação em inversão fraca (*subthreshold*).

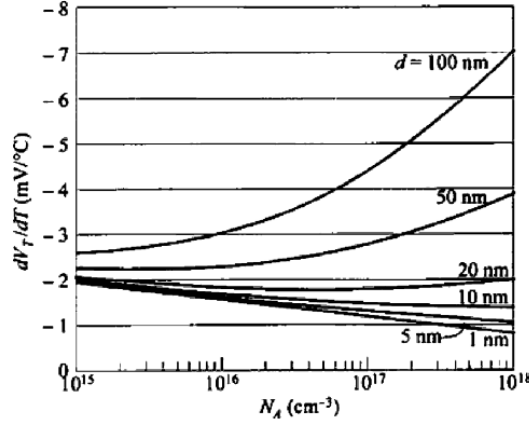


Figura 2.4: Alteração da dependência da temperatura de  $V_{TH}$  pela espessura do óxido  $d$  e dopagem  $N_A$  [2].

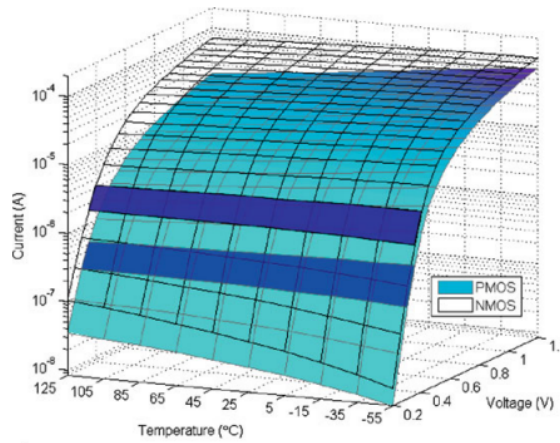


Figura 2.5: Dependência da corrente, devido a variações PVT numa tecnologia de 90 nm [2].

$$I_D = \mu_n C_{ox} \frac{W}{L} \left( (V_{GS} - V_{TH}) V_{DS} - \frac{V_{DS}^2}{2} \right), \quad V_{GS} > V_{TH} \quad \text{e} \quad V_{DS} < V_{GS} - V_{TH} \quad (2.1)$$

$$I_D = \frac{\mu_n C_{ox}}{2} \frac{W}{L} (V_{GS} - V_{TH})^2 [1 + \lambda (V_{DS} - V_{DSat})], \quad V_{GS} > V_{TH} \quad \text{e} \quad V_{DS} \geq V_{GS} - V_{TH} \quad (2.2)$$

O valor da tensão de *threshold* varia com a temperatura (equação 2.4) por efeito da sensibilidade do potencial de Fermi à temperatura (equação 2.5).

$$V_{TH} = V_{FB} + 2\phi_F + \gamma \sqrt{2\phi_F} \quad (2.3)$$

$$\frac{\partial V_{TH}}{\partial T} = \frac{\partial \phi_{gs}}{\partial T} + 2 \frac{\partial \phi_F}{\partial T} + \frac{\gamma}{\sqrt{2\phi_F}} \frac{\partial \phi_F}{\partial T} \quad (2.4)$$

$$\phi_F = (kT/q) \ln(N_A/N_I) \quad (2.5)$$

Um exemplo da variação da derivada da  $V_{TH}$  com a temperatura para diferentes valores da espessura do óxido  $d$  e da dopagem  $N_A$  é apresentado na figura 2.4.

Atendendo a equação 2.2, além da variação da temperatura pela tensão de *threshold* ( $V_{TH}$ ) há também efeitos nesta variação pela mobilidade eletrônica ( $\mu$ ). A redução da mobilidade com o aumento da temperatura é dado por

$$\mu(T) = \mu(T_0) \cdot \left(\frac{T_0}{T}\right)^{1,5} \quad (2.6)$$

onde  $T_0$  é um valor de temperatura de referência e  $T$  a variação de temperatura [21].

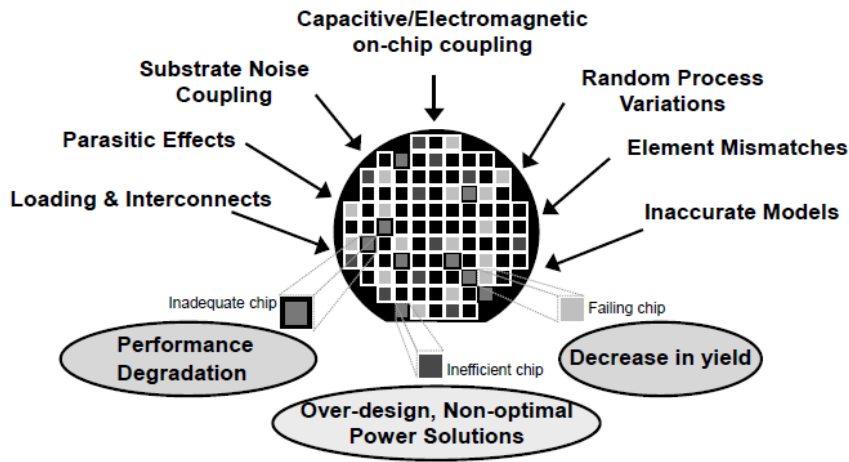


Figura 2.6: Contribuições na variabilidade do fabrico de CMOS em escala nanométrica [3].

Dentro dos efeitos PVT, os mais críticos tendem a ser os devidos à variabilidade do processo. A variabilidade do processo é geralmente tida como um novo desafio atendendo à diminuição do tamanho da tecnologia. No entanto, os problemas associados a estes desvios já eram considerado há mais de 50 anos quando Shockley fez um estudo da flutuação aleatória em junções p-n [22]. Atualmente estes desvios são cada vez mais considerados com o uso de tecnologias nanométricas em CMOS, tendo um forte impacto no desempenho do circuito, nomeadamente em termos de maiores tempos de propagação e aumento dos consumos dinâmico e estático. Estes podem implicar a degradação do processo de fabrico (*yield*), levando a custos acrescidos e a maior tempo de produção.

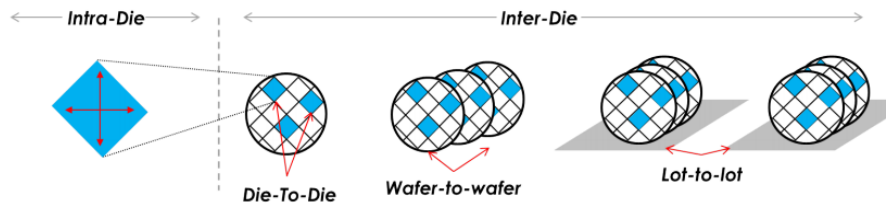


Figura 2.7: Classificação das variações de processo [4].

Estas variações são agrupadas, do ponto de vista do momento em que ocorrem e da extensão dos seus efeitos, como variações 2.7:

- *Intra-die* — resultantes de limitações/efeitos estocásticos do equipamento durante o processo de fabrico de um circuito específico (*die*) decorrentes, por exemplo, de variações associadas ao processo CMP (planarização químico-mecânica) e de efeitos de alinhamento e proximidade ótica (limitação da lente). Por forma a caracterizar estas variações de processo, é usado o método estatístico de Monte Carlo;
- *Inter-die* — constantes em todo o *die* variando entre diferentes *die* de uma *wafer* ou lote de *wafers*. Estas variações são geralmente cobertas em simulações de cantos (*corners*);

Na indústria, a designação mais usada para as variações de processo é associada às situações mais extremas denominadas de cantos (*corners*) e identificadas por duas letras. Nestas, a primeira refere-se a variações nos transístores NMOS e a segunda nos transístores PMOS. As quatro combinações mais comuns são: FF (*fast-fast*), SF (*slow-fast*), SS (*slow-slow*), FS (*fast-slow*) e TT (*typical-typical*); que designam combinações de valores dos parâmetros que levam a um desempenho mais rápido (F) ou mais lento (S) dos transístores em portas lógicas, comparando com o comportamento nominal esperado (TT) de cada um dos transístores [23].

### 2.1.1 Efeitos PVT numa PLL

Estes efeitos PVT implicam falhas e deterioração do desempenho de uma PLL, sendo a sua própria dinâmica e funcionamento alterados em relação ao comportamento nominal pretendido.

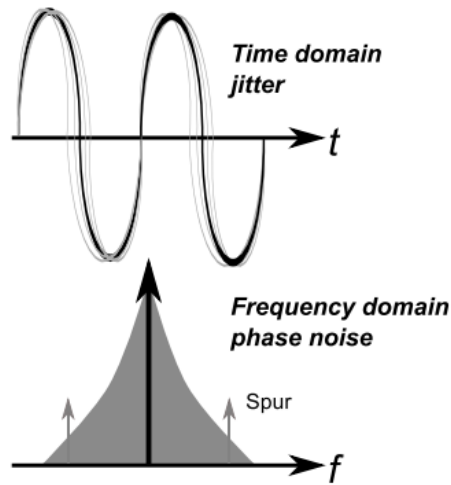


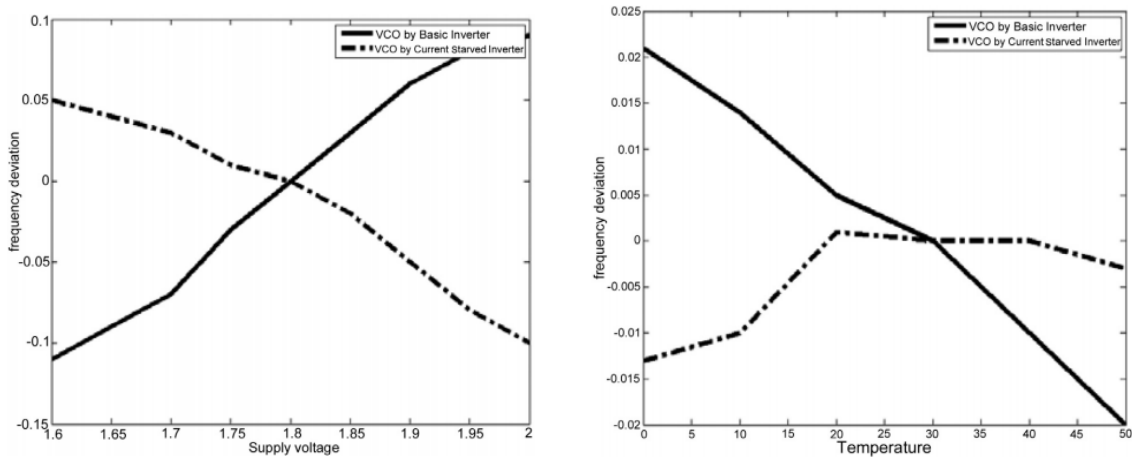
Figura 2.8: Alterações no desempenho da PLL por efeitos PVT [3].

A observabilidade dos efeitos PVT é geralmente categorizada, no que diz respeito ao desempenho da PLL, em três grandes parâmetros [24] :

- Ruído de fase;
- Espúrios;
- Tempo de aquisição e *overshoot*.

A implicação destas interferências levam a que parâmetros fundamentais para o correto funcionamento do sistema, sejam modificados, podendo alterar o sinal de tal modo que ele saia da *Capture-Range*, impedindo assim que a PLL nunca consiga fazer *Lock* ao sinal de entrada.

e podem ser de tal modo extremos, a ponto de levar a PLL a sair da gama de captura e impedindo que consiga fazer sintonia (*Lock*) com o sinal de entrada.



(a) Variações de tensão.

(b) Variações de temperatura.

Figura 2.9: Impacto de variações VT na frequência à saída do VCO [5].

Cada bloco da PLL apresenta uma sensibilidade própria a variações PVT e a de cada um deles depende do modo como for implementado. Na figura 2.9 ilustra-se o impacto das variações de tensão e de temperatura na frequência de VCO do tipo oscilador em anel implementados com inversores simples (alimentados em tensão) e com inversores alimentados em corrente (current starved).

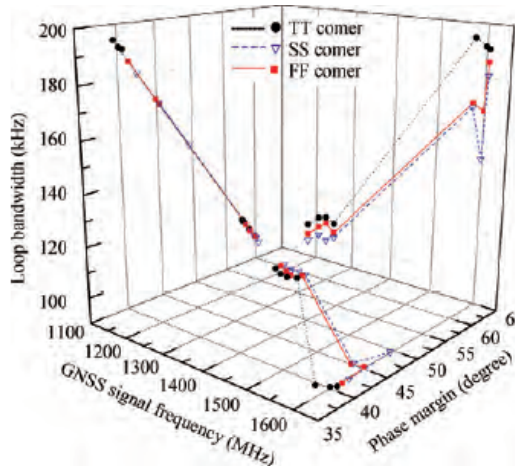


Figura 2.10: Impacto de variações PVT na banda de passagem em malha fechada de uma PLL [6].

Na figura 2.10 ilustra-se o impacto de variações do processo na largura de banda do ganho em anel e na margem de fase, a diferentes frequências do sinal de sistemas de navegação global por satélite (GNSS).

## 2.2 Métodos de teste - Aplicações e BIST

Dadas as imperfeições resultantes dos efeitos PVT, é comum recorrer a mecanismos de correção do funcionamento do circuito. Um conjunto significativo destas soluções recorre a circuitos auxiliares que promovem correções através de valores e cálculos feitos em tempo real [25].

### 2.2.1 Circuitos auxiliares de teste

Os circuitos auxiliares de teste baseiam-se em métodos de *built-in self-test* (BIST). Estes permitem que um sistema se autoteste de modo a manter requisitos tais como alto rendimento de produção, reduzida necessidade de ciclos de reparação, e evitar as desvantagens associadas a limitação de acesso a nós de teste e a maior custo de teste durante o processo de fabrico. O propósito principal dos mecanismos de BIST é o de ultrapassar a complexidade e como tal reduzir o custo resultante da utilização de equipamento de teste externo.

A figura 2.11 ilustra um caso de aplicação de mecanismos de BIST por forma a corrigir e testar variações de entradas no sistema. É feita uma comparação com valores já conhecidos e aplicar estímulos que são observáveis à saída.

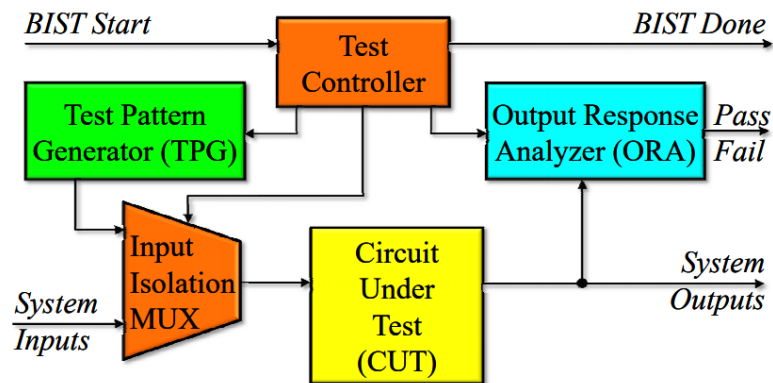


Figura 2.11: Exemplo de diagrama de blocos de um mecanismo de *BIST* [7]

### 2.2.2 Redes de sensores e DSP

Em [3] a realização de testes com circuitos auxiliares é feita recorrendo a uma rede de sensores que observam diferentes grandezas ou parâmetros, com os quais se permite saber o estado do num sistema de comunicação rádio. Dado o recurso a circuitos de processamento digital de sinal (DSP), como ilustrado na figura 2.12, é requerida a conversão dos sinais observados em valores digitais.

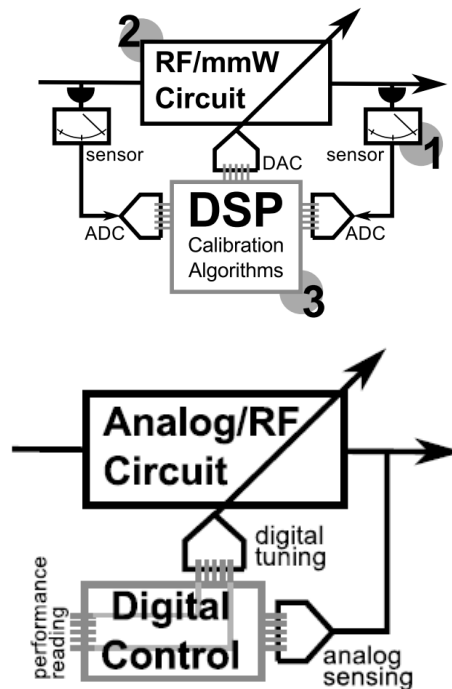


Figura 2.12: *Sensor network* usando a um DSP para calibração de um circuito RF [3]



Esta abordagem implica em geral grandes custos, dada a utilização de diversos sensores e o DSP ser geralmente de grandes dimensões. Outra desvantagem é a necessidade de cálculo em tempo real, exigindo que o potencial de cálculo do DSP seja suficientemente elevado para manter os níveis desejados de desempenho do circuito RF.

### 2.2.3 Aplicação de *PSO* para reconfiguração automática de *Hardware*

O método de *Particle Swarming Optimization* procura a melhor solução de um problema segundo uma abordagem de aproximações sucessivas, em que em cada iteração se experimenta nova combinação dos dados disponíveis, no sentido de otimizar a solução candidata até que seja satisfeita uma medida de qualidade pré-estabelecida. Esta abordagem recorre a padrões conhecidos que, através de métodos heurísticos, definem qual a seleção de *features* fundamentais por forma a melhorar as características do sistema.

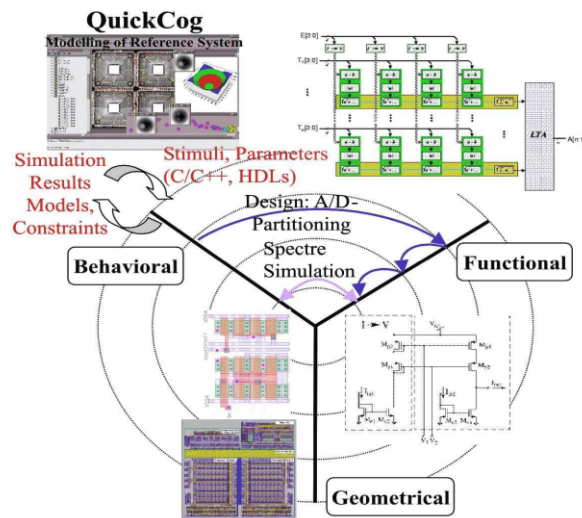


Figura 2.13: Metodologia para conceção em sistemas inteligentes de baixa potência reconfiguráveis [8].

Em [8] é apresentada uma abordagem baseada em PSO com o objetivo de aumentar a tolerância a desvios de desempenho de circuitos sensores de muito baixo consumo. Para o efeito são usados circuitos de baixa potência reconfiguráveis, compostos por uma matriz de células (Fig.2.15). Diferentes combinações destas células permitem obter fatores de correção dos desvios detetados, tomando um modelo comportamental e a definição da função objetivo do algoritmo *PSO*. A determinação da melhor combinação a usar em cada caso é baseada em classificadores 1-NN (*nearest neighbor*). Nos métodos baseados em classificadores k-NN é atribuída uma classificação aos dados de treino k mais próximos no espaço total de dados (*feature space*), cujo resultado da classificação é o objeto considerado classificado, isto é, com o objeto sendo atribuído à classe mais comum entre seus k vizinhos mais próximos — para k positivo e inteiro, tipicamente pequeno — [26].

O algoritmo de *PSO* utilizado, resumido na figura 2.14, consiste em, partindo de um grande conjunto de dados de treino (*train samples*), dividi-los em conjuntos de treino e de teste. Dos dados de treino são extraídos os melhores conjunto de parâmetros (*features*) característicos do desempenho dos sensores usados para classificar os dados de teste. A escolha das melhores *features* é um desafio nestas implementações, decorrendo daí a necessidade de aplicar algoritmos de *feature selection*, os quais irão ser descritos na secção 2.3.1.

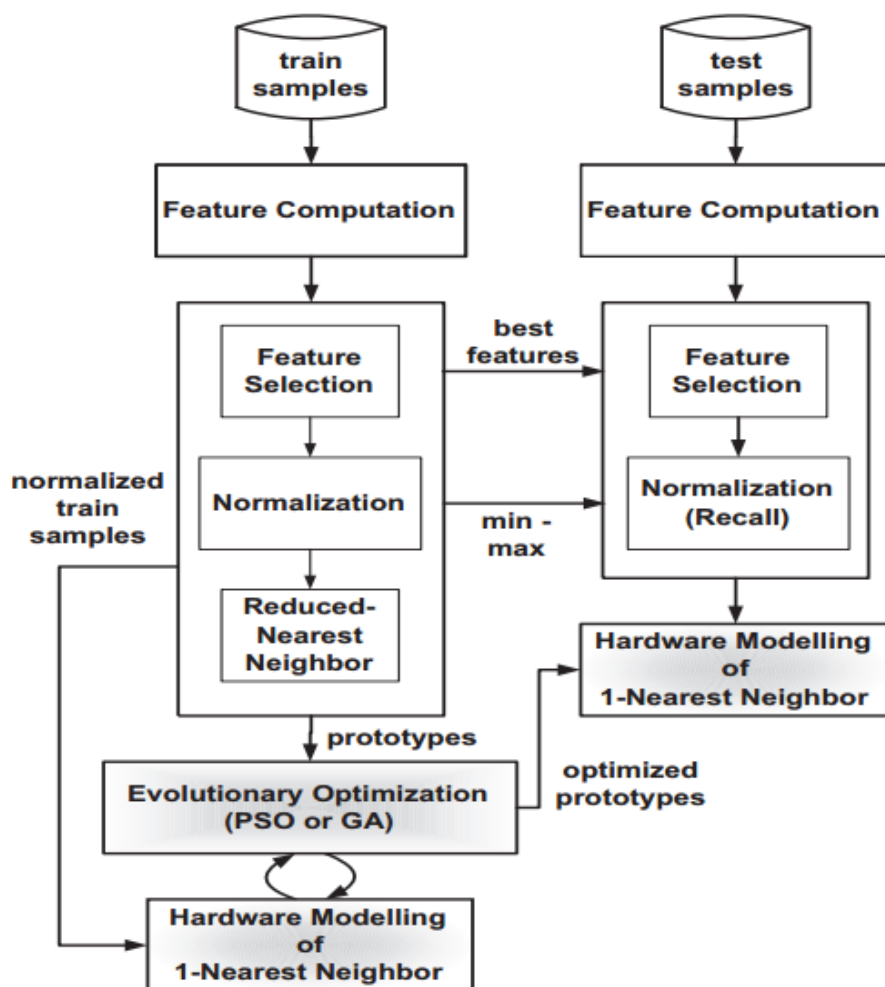


Figura 2.14: Algoritmo usado em *PSO* [8].

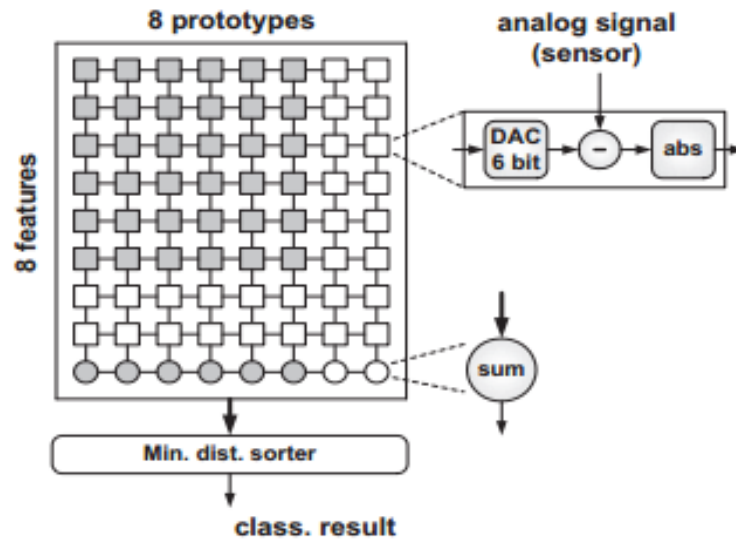


Figura 2.15: Diagrama de blocos de um modelo de classificação 1-NN [8].

Segundo os resultados apresentados neste artigo consegue-se um ganho de 67% no número de circuitos corretamente classificados, o que seria equivalente a um aumento do rendimento da produção de 67%.

#### 2.2.4 Cancelamento de efeitos PVT usando sistemas de autoteste

No artigo [9] é apresentado um método de compensação de efeitos PVT em PLL baseado na utilização do relógio de referência externo e num FDC (*frequency-to-digital converter*) usado para converter as frequências dos sinais de referência e de saída da PLL em valores numéricos. Através da contagem do número de ciclos de relógio à saída do oscilador é possível decidir (em tempo-real), após comparação com uma referência, se estes correspondem a um aumento/decréscimo do valor pretendido da frequência de funcionamento em *steady-state*.

Inicialmente, para compreender as falhas no desempenho da PLL, é feita uma análise de sensibilidades juntamente com simulações de cantos (*corners*) críticos para diferentes processos e temperaturas, por forma a pesar os efeitos PVT. A figura 2.16 ilustra a variação do *jitter* em consequência do ruído gerado na tensão de alimentação, em função do aumento da corrente média do VCO e da capacidade de desacoplamento da tensão de alimentação.

O método proposto foi testado na implementação de uma *charge-pump* PLL, mostrando uma compensação adequada de PVT melhorando o desempenho em *jitter* e consumo de energia como resultado.

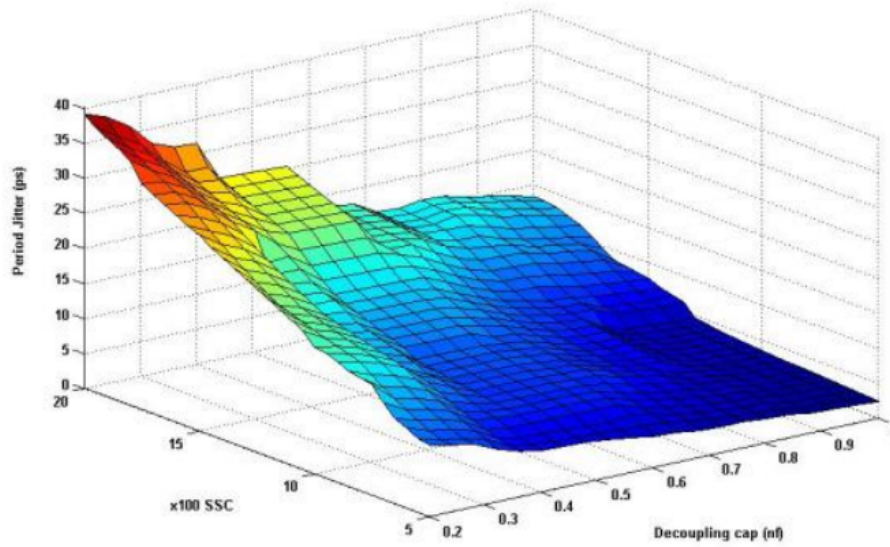


Figura 2.16: Dependência do *jitter* em função da corrente média de comutação e da capacidade de desacoplamento da fonte de alimentação [9].

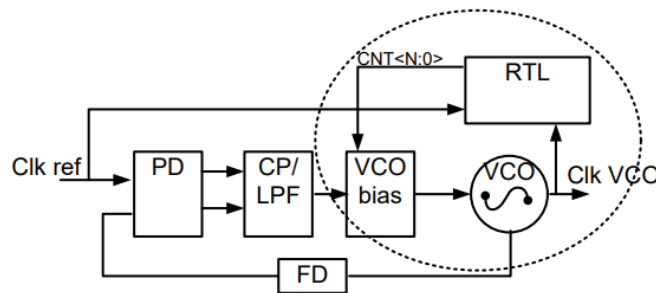
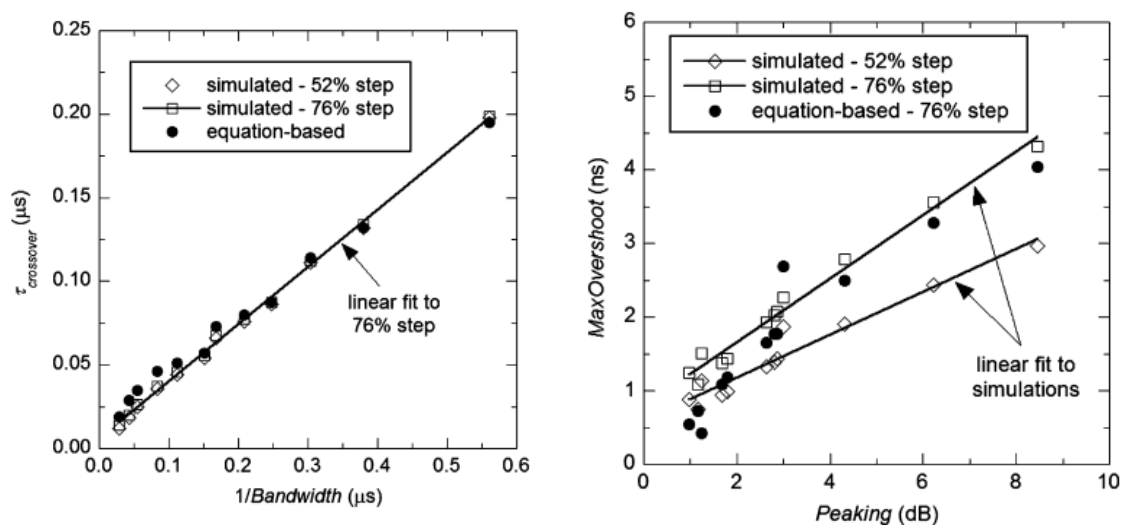


Figura 2.17: Circuito proposto para compensação e detecção PVT [9].

Como ilustrado na figura 2.17 a calibração é feita ajustando a polarização do VCO. A palavra  $CNT\langle N:0 \rangle$  permite alterar o comportamento do VCO de modo a diminuir os efeitos do *jitter* apresentados na figura 2.16, forçando-o o VCO a seguir um valor ótimo para definir o ponto correto de oscilação.

Como referido em 2.1.1, para além das variações no sinal à saída, é feito no artigo [10] uma observação diferente da apresentada por [9], observando os efeitos PVT na resposta dinâmica, mais concretamente na resposta ao degrau de frequência resultante da alteração na reprogramação do bloco do divisor N. É apresentado um circuito digital de autoteste baseado numa máquina de estados, que altera o bloco divisor da frequência, apresentando um reencaminhamento alternativo do sinal à saída do VCO numa malha com valor de N distinto. Dada a dependência de parâmetros como *overshoot* e a banda da PLL com o valor de N, é possível caracterizar variações PVT com comparação de valores que seguem a solução de uma equação associada a estes 2.18.



(a) Variação da largura de banda na resposta ao degrau a duas frequência distintas. (b) Variação do *overshoot* na resposta ao degrau a duas frequência distintas.

Figura 2.18: Resultados do mecanismo de autoteste apresentado em [10].

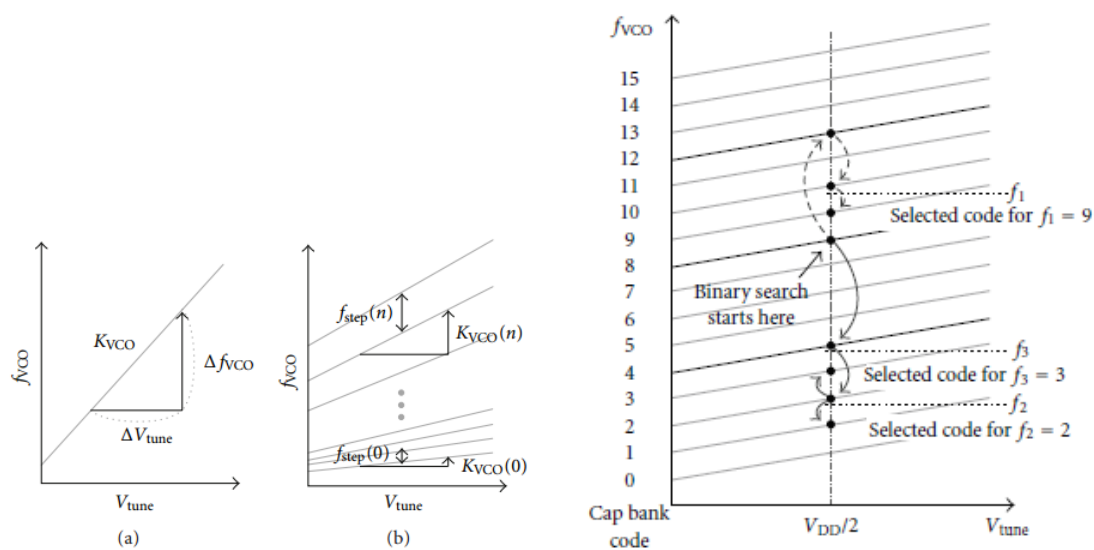
Por forma a avaliar a capacidade de medição do sistema é apresentado na figura 2.18 uma comparação dos valores simulados com os estimados. O recurso a este tipo de estimações apresenta uma diminuição do tempo/custo associado a testes de produção — em *wafer-level* e *package* — aumentado assim a cobertura do desenho para teste (*design-for-test*). No entanto, apesar das considerações feitas não é demonstrado uma forma de criar um sistema que permita uma autocalibração da PLL, sendo feito apenas uma demonstração do exercício da *testbench* aplicada.

### 2.2.5 Considerações de autocalibração para sintetizadores de frequência

Em [11] é apresentada uma metodologia de avaliação do funcionamento do oscilador da PLL. A ideia apresentada neste artigo consiste em encontrar um ponto de funcionamento inicial ideal, que optimize o bloco em causa. A forma como é executado o algoritmo proposto recorre a uma recursividade não parametrizada, isto é, sem limites superiores/inferiores previamente estabelecidos, por forma a comparar os valores de  $K_{vco}$  com aqueles que a PLL apresenta um melhor funcionamento.

Dada a dependência do  $K_{vco}$  com os valores das capacidades, seja a implementação feita usando um oscilador LC ou em osciladores em anel, a correção é feita por variação do valor da capacidade, usando para isso um banco de capacidades.

A forma como é feita a escolha do valor a usar no banco de capacidades consiste na leitura e comparação direta com a frequência de referência, como é ilustrado na figura 2.20. A leitura da frequência de oscilação é feita forçando o oscilador a operar num ponto de funcionamento determinado, neste caso  $V_{dd}/2$ , desligando-o do resto do circuito, sendo portanto uma calibração feita (*off-line*). É feita aqui uma pesquisa binária (*binary search algorithm*), que consiste numa busca do conjunto do código C — à saída do banco de capacidades da figura 2.20 — que faça



(a)  $K_{VCO}$  para diferentes valores do código do banco de capacidades.

(b) Pesquisa binária do valor de  $K_{VCO}$ .

Figura 2.19: Metodologia principal proposta em [11].

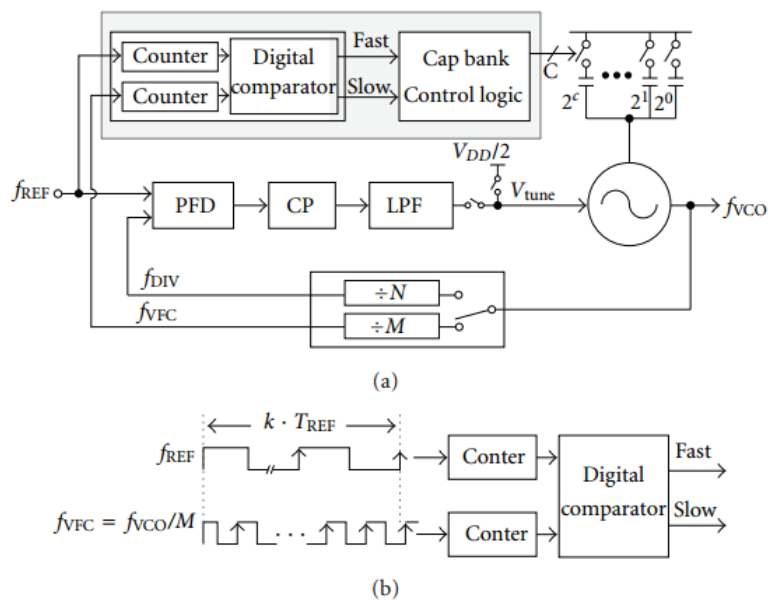


Figura 2.20: Comparação do VCO relativamente a uma frequência de referência. (a) Circuito proposto. (b) Princípio de operação [11].

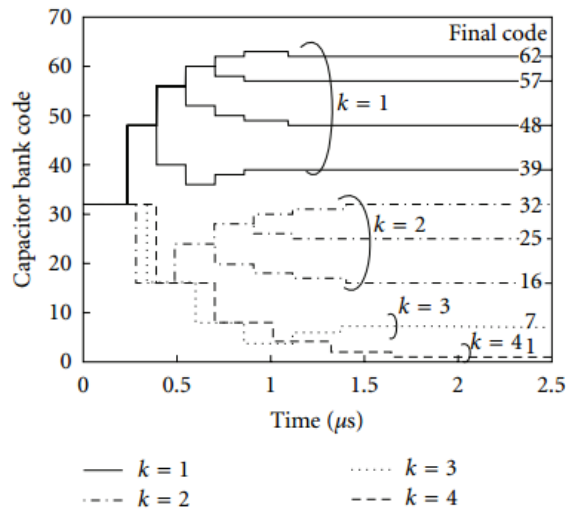


Figura 2.21: Convergência do princípio de autocalibração por meio de comutação de capacidades [11].

convergir para a solução, representado numa comutação sequencial do bit correspondente a cada *switch* que ativa uma capacidade, por forma a que os valores de *fast* e *slow* do comparador digital atinjam um *threshold* mínimo.

O valor máximo do conjunto total do banco de capacidades é apresentado com um resolução direta de 16bits, não sendo apresentado um estudo prévio de como é obtido, tomando-se como uma gama "suficiente" para cobrir todos os casos de desvios, é portanto dada sem colocar nenhum estudo PVT anterior, sendo que daí, o valor do banco de capacidades conter um valor arbitrário inicial, sem qualquer limite.

Este tipo de abordagem para mecanismos de autocalibração, baseados em comutação de um conjunto de elementos que alteram a dinâmica de um bloco da PLL, tem sido muito utilizado na literatura atual, sendo que apresenta uma

No artigo [12] é apresentada uma metodologia a qual segue a comutação de um banco de resistências, com recurso a blocos digitais — contadores, comparadores, etc — por forma a garantir medir desvios das variações PVT, e calibrar a PLL, observando as variações de *jitter* à saída do oscilador.

Na figura 2.22 os autores sugerem uma comutação de um banco de resistências, por forma a variar a corrente num oscilador LC, esta que afeta diretamente a frequência de oscilação deste, sendo feita uma deteção dos desvios desta (*jitter*) com recurso a um circuito de deteção de picos e contagem. A convergência desta solução para o resultado pretendido deste é muito semelhante ao apresentado em [11].

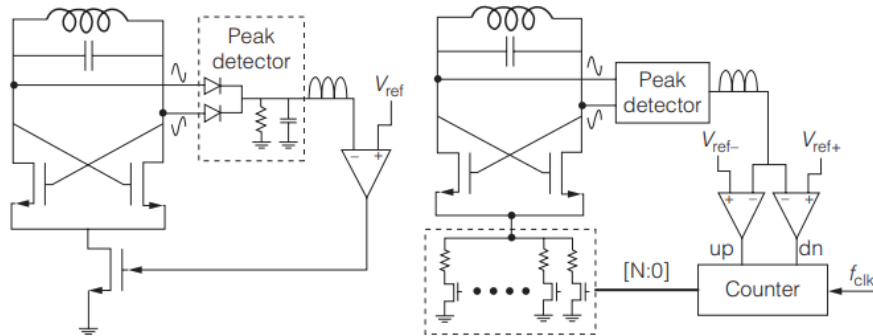


Figura 2.22: Autocalibração de um oscilador LC [12].

## 2.3 Algoritmos de seleção de variáveis e reconhecimento de padrões

A aprendizagem automática (em inglês: "machine learning") é um subdomínio da ciência da computação que evoluiu do estudo do reconhecimento de padrões e da teoria da aprendizagem computacional em inteligência artificial. Em 1959, Arthur Samuel definiu aprendizagem automática como o "campo de estudo que dá aos computadores a facilidade de aprender sem serem explicitamente programados". A aprendizagem automática explora o estudo e construção de algoritmos que podem aprender com os próprios erros e fazer previsões sobre dados. Tais algoritmos operam construindo um modelo a partir da amostragem das entradas, a fim de fazer previsões ou decisões guiadas pelos dados ao invés de simplesmente seguir instruções programadas inflexíveis e estáticas. Enquanto que na inteligência artificial existem dois tipos de raciocínio (o indutivo, que extrai regras e padrões de grandes conjuntos de dados, e o dedutivo, que recorre a uma análise lógica para construir argumentos, através de premissas/argumentos para obter uma conclusão), a aprendizagem automática só se preocupa com o indutivo.

Dentro do mundo de aprendizagem automática existe um grande conjunto de algoritmos que, como referido em [27, 28], é necessário selecionar criteriosamente de modo a melhor adequar à aplicação em causa, tendo também presente o objetivo final.

A aprendizagem automática pode ser classificada em duas grandes categorias:

- Aprendizagem supervisionada: a máquina aprende através de um conjunto de saídas e entradas, de modo a mapear uma regra de conexão entre elas usando um mediador, também usado o termo "professor" [29].
- Aprendizagem não supervisionada: nestes não existe uma regra a seguir uma vez que o algoritmo aprende sozinho através das entradas e o resultado destas na saída; são métodos muito exigentes mas que permitem chegar a padrões mais difíceis de obter por mediação.

As aplicações podem ser categorizadas pelo impacto que têm no conjunto de dados subjacente. Sendo dadas por :

- Classificação;



- Regressão;
- Agrupamento (*Clustering*);
- Estimação de densidade para distribuições;
- Redução do espaço/dimensão de dados, também conhecido por *feature selection*.

Esta última apresenta características muito interessantes no que diz respeito à integração para uso em circuitos complexos onde é difícil encontrar uma (boa) metodologia para decidir quais os fatores que realmente intervêm no sistema [30, 8]. As aplicações como a regressão apresentam também uma integração apropriada em sistemas eletrônicos, de modo a que permite prever e definir padrões utilizando grandes conjuntos de dados [31, 8, 14]. O método apresentado na subsecção 2.2.3, é um método de aprendizagem supervisionada, sendo que o impacto no conjunto de dados consiste em uma classificação destes.

### 2.3.1 Feature selection

Por *feature*, no contexto de aprendizagem automática e reconhecimento de padrões, entende-se uma característica associada a um fenómeno a ser observado e/ou uma propriedade mensurável [32]. As *features* são representadas geralmente por valores numéricos, no entanto existem também *features* estruturais como palavras e grafos geralmente usados em reconhecimentos de padrões e em SVM. O conceito de *feature* é relacionado como o seguimento de uma variável atributiva usada em métodos estatísticos.

A definição de *feature selection* pode resumir-se em, dado um conjunto de *features*  $X = \{x_i | i = 1..N\}$ , encontrar o subconjunto  $Y_M$ , com  $M < N$ , que maximize uma função objetivo  $j(Y)$ , idealmente  $P(\text{correct})$  apresentado em [13, 33].

$$Y_M = \{x_{i1}, x_{i2}, \dots, x_{iM}\} = \arg_{M, i_M} \max J \{x_i | i = 1..N\} \quad (2.7)$$

Os dois métodos apresentados nesta secção baseiam-se na forma de aproximação que se deseja ter, sendo mostrados os seguintes algoritmos.

- *Sequential forward selection*

Começando com um conjunto vazio, sequencialmente adicionar a *feature*  $x^+$  que maximize  $j\{Y_K + x^+\}$  quando combinadas com as *features*  $Y_k$  que já foram previamente selecionadas no algoritmo 1.

---

#### Algorithm 1 Sequential forward selection

---

- 1: Começa com um conjunto vazio  $Y_0 = \{\emptyset\}$
  - 2: Seleciona a melhor *feature* seguinte  $x^+ = \arg_{x \notin Y_k} \max J(Y_k + x)$
  - 3: Atualiza  $Y_{k+1} = Y_k + x^+$ ;  $k = k + 1$
  - 4: Vai para o 2º estado
-

Este algoritmo tem melhores resultados quando o subconjunto de *features* ótimo é pequeno. Quando a procura ainda está no início, isto é o conjunto está vazio, um grande número de estados pode ser avaliado. Se o conjunto estiver quase a completar-se, a região examinada pelo algoritmo 1 é mais estreita sendo que a maioria das *features* já foi selecionada. A maior desvantagem do algoritmo 1 é que é incapaz de remover *features* consideradas obsoletas após a adição de outras *features*.

- *Sequential backward selection*

Começando por um conjunto completo, remove-se sequencialmente a *feature*  $x^-$  que menos reduz o valor da função objetivo  $J(Y - x^-)$ . Neste tipo de funções, conhecidas por *non-monotonic*, apesar de se lhe remover *features*, aumenta a função objetivo, isto é,  $J(Y_k - x^-) > J(Y_k)$ .

---

**Algorithm 2** *Sequential backward selection*

---

- 1: Começa com um conjunto completo  $Y_0 = X$
  - 2: Remove a pior *feature*  $x^- = \arg_{x \in Y_k} \max J(Y_k - x)$
  - 3: Atualiza  $Y_{k+1} = Y_k - x^-$ ;  $k = k + 1$
  - 4: Vai para o 2º estado
- 

O algoritmo 2 funciona melhor quando o subconjunto de *features* ótimo é grande, dado que irá ocupar mais tempo a selecionar grandes subconjuntos. A maior limitação deste algoritmo é a incapacidade de reavaliar a qualidade de uma *feature* previamente descartada [34].

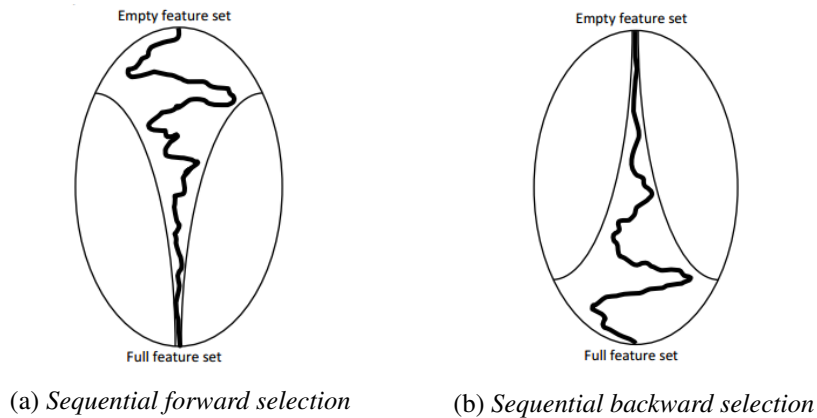


Figura 2.23: Comparação gráfica do *feature set*, sendo a região mais estreita a representação da evolução dos algoritmos [13].

Têm sido propostos vários métodos e algoritmos de *feature selection* [35, 36]. A aplicação destes em circuitos eletrónicos analógicos e mistos é então um desafio que se justifica dadas as vantagens inerentes, sendo ainda relativamente reduzido o número de casos de aplicação conhecidos. Apresenta-se de seguida o exemplo de aplicação a um sistema microeletromecânico.

### 2.3.2 Redução do erro de frequência em MEMS

Nesta subsecção é apresentado um exemplo de aplicação de um algoritmo *Sequential backward-forward selection* no projeto de um oscilador microeletromecânicos (MEMS) de modo a minimizar a sensibilidade da sua frequência de oscilação a variações estruturais ocorridas no processo de fabrico [14].

Os sistemas MEMS, também referidos como micromáquinas, são sistemas “inteligentes” construídos à escala micrométrica, ou mesmo nanométrica, que envolvem estruturas mecânicas e circuitos eletrónicos, implementados sobre um mesmo substrato de silício. Em alguns casos, podem também encontrar-se sistemas óticos e mesmo químicos.

Os MEMS osciladores ou ressonantes, têm sido considerados como substitutos para os osciladores baseados em cristal de quartzo em sistemas eletrónicos. Um oscilador MEMS, comparando com o cristal de quartzo, é mais pequeno, mais barato e tem compatibilidade com CMOS. Quanto a comparação do consumo de potência é ainda um tema de discussão, dada a variabilidade de aplicações destes [37]. Deste modo os MEMS são potenciais candidatos a serem usados em aplicações em circuitos eletrónicos de pequena escala.

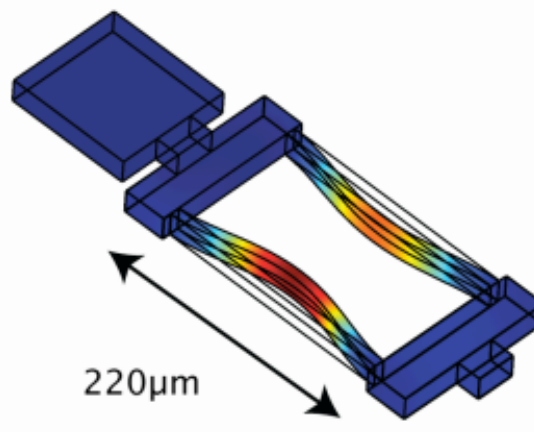


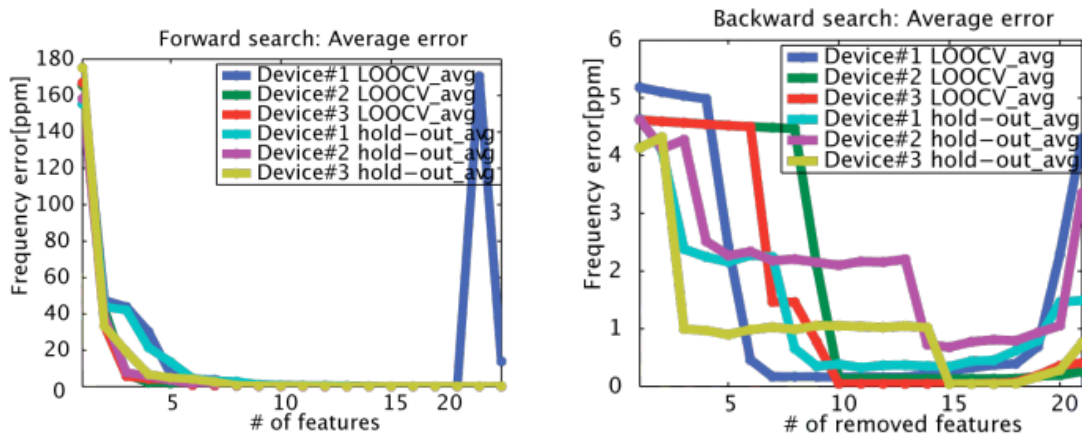
Figura 2.24: Desenho de um oscilador MEMS [14]

Contudo, como as dimensões dos MEMS são reduzidas as interferências das PVT afetam mais estes do que aos cristais de quartzo. No estudo referido em [14] o processo de otimização é feito com recurso a algoritmos de aprendizagem automática, por forma a diminuir o número de pontos para calibrar e testar o circuito, mantendo um nível aceitável de erro. Começa-se por adicionar/remover — dependendo do conjunto inicial de *features* — as *features* que alteram a equação de oscilação da MEMS, e alterando estas, é avaliada consoante um valor objetivo, se a inserção/remoção destas permitem convergir para uma solução aceitável. Dado que a frequência de oscilação depende da tensão de polarização  $V_{bias}$  e da temperatura, é necessário construir uma tabela que relaciona a temperatura com a  $V_{bias}$  para um valor fixo de frequência, isto é  $f_{osc} = f(V_{bias}, T_{temp})$ . Definindo a curva que mais se ajuste ao comportamento da frequência de saída da MEMS em 2.8, tomando  $\{x^{(i)}, f^{(i)} : i = 1, 2, \dots, m\}$  — em que  $m$  corresponde ao número de dados

de treino —, onde  $x^i = [x_1^{(i)}, x_2^{(i)}]^T$  para o par temperatura- $V_{bias}$  e  $f^{(i)}$  a frequência medida.

$$f^{(i)} = \sum_j \sum_k a_{j,k} \left(x_1^{(i)}\right)^j \left(x_2^{(i)}\right)^k \quad \text{onde, } 0 \leq j, 0 \leq k, i+j \leq n \quad (2.8)$$

Verificando diferentes dispositivos, de modo a comparar as diferenças no seu processo, e aplicando a tabela é possível comparar o efeito das interferências PVT na calibração do oscilador MEMS. Para determinar o melhor funcionamento e correção do erro associado à frequência desejada, é aplicado o algoritmo de *sequential backward-forward selection* que seleciona os valores corretos do par temperatura- $V_{bias}$  para estabilizar o sistema. A figura 2.25 demonstra a eficácia da aplicação dos algoritmos referenciados. Pode ver-se que ao fazer uma seleção do conjunto de fatores que alteram a dinâmica da oscilação da MEMS, é possível obter um melhoramento do desempenho do sistema, sem recorrer a uma complexidade superior, caso se for considerado todos as *features* para calibração.



(a) Erro médio usando *Sequential forward selection* (b) Erro médio usando *Sequential backward selection*

Figura 2.25: Redução do erro da frequência aplicando algoritmos de *Sequential backward-forward selection* [14].

A inclusão de circuitos auxiliares de teste e de calibração em circuitos eletrônicos, nomeadamente no analógicos e mistos, permite uma mais eficiente deteção de falhas e a correção do seu desempenho, evitando assim custos adicionais com perdas de circuitos defeituosos, após o fabrico e durante o seu serviço na aplicação.

Porém, o projeto destes circuitos auxiliares nem sempre segue metodologias de otimização, nomeadamente porque não é avaliado corretamente todo o domínio das variações em causa e não são usados os modelos que mais fielmente tratam essas variações e dependências. Este facto pode ter consequências no dimensionamento do número de células para efeitos da calibração, em que não é estabelecido uma correlação dos valores máximo necessários para satisfazer uma maior cobertura de efeitos PVT, sem o sobre-dimensionamento — como é o caso do dimensionamento

dos bancos de capacidade apresentados — levando a que exista um desperdício a nível de consumo e ocupação em área por circuitos que podem nem ser utilizados.

A aplicação de ferramentas de *machine learning* permite desenvolver mecanismos mais eficientes de calibração, sendo que o uso destas permite uma melhor seleção e previsão de tratamentos de dados.

Nesta dissertação, recorre-se ao uso destes por forma a caracterizar o impacto de efeitos PVT no desempenho de circuitos analógicos-mistos, mais concretamente uma PLL.



## Capítulo 3

# Fundamentos de desenho da PLL

A *PLL* é um circuito misto que se baseia numa malha de realimentação de fase para manter a frequência do sinal gerado sincronizada com a frequência do sinal de entrada tomado como referência. A figura 3.1 ilustra o diagrama de blocos genérico de uma PLL — o sinal de erro  $e(t)$  resultante da comparação das fases dos sinais de entrada ( $ref(t)$ ) e realimentado ( $div(t)$ ) constitui, após filtragem para extrair o seu valor médio  $v(t)$ , o sinal de controlo de um oscilador controlado por tensão (VCO) de modo a que este adapte a sua frequência à frequência do sinal de entrada. A frequência do sinal de saída  $out(t)$  é  $N$  vezes maior do que a do sinal de entrada caso na malha de realimentação seja usado um divisor por  $N$  [15].

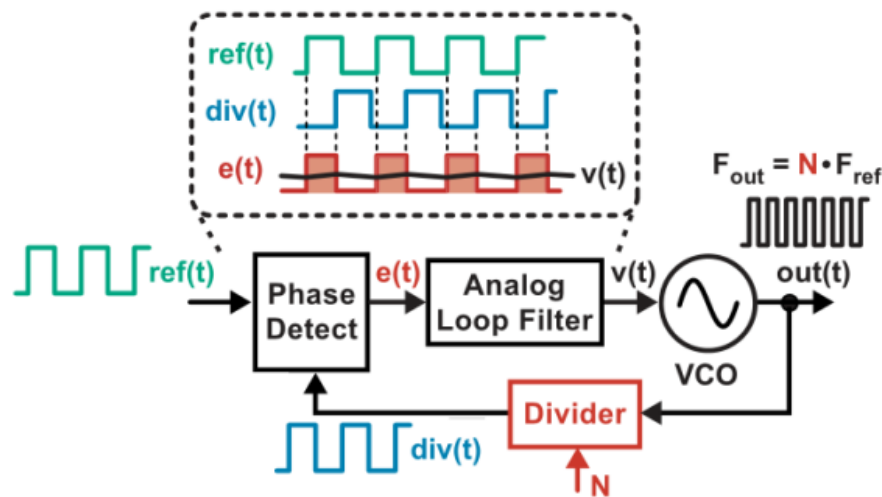


Figura 3.1: Arquitetura de uma PLL [15].

Nas primeiras implementações de PLL, os blocos constituintes eram inteiramente analógicos: um multiplicador analógico como detetor de fase, um filtro *RC* passivo e/ou ativo do tipo passa baixo, e o bloco do VCO como o gerador o sinal de saída. Mais recentemente, com a evolução dos circuitos digitais, o uso de detetores de fase digitais tornou-se mais recorrente, sendo estes geralmente portas em disjunção exclusiva (*XOR*) ou multivibrador biestável (*flip-flops*) do tipo JK e D. Este tipo de PLL é denominada digital, ou híbrida/misto.

É também possível desenhar PLL puramente digitais (ADPLL), como é, por exemplo, proposto em [38]. Para efeitos de emulação, a PLL pode ser modelada recorrendo a modelos dos próprios circuitos constituintes ou a modelos comportamentais que simulam apenas o comportamento linearizado dado pelas funções de transferência dos blocos. A título de exemplo, é implementado uma Software PLL (SPLL), através da sua emulação numa DSP em [39].

Para efeitos de classificação, a ordem da PLL é determinada pelo número de polos da função de transferência, e o tipo dado pelo número de polos na origem (ou integradores) da função de transferência em malha aberta. Como o VCO contribui sempre com um polo na origem, carácter inerente a integração de uma tensão/corrente de controlo, cada PLL é pelo menos da ordem 1 e tipo 1. Com a contribuição dos polos do filtro a ordem e o tipo de PLL aumentam e quanto maior o tipo, melhor é o desempenho da PLL em acompanhar a frequência e a fase do sinal de entrada. Por exemplo, uma PLL de tipo 2 é capaz de seguir um salto em fase, bem como em frequência com erro zero de fase no estado estacionário, enquanto que uma PLL do tipo 1 só consegue acompanhar um mudança de salto em fase.

São de seguidas apresentadas PLL baseadas em circuito mistos tipo 2 de ordem 3, uma vez que as deduções e princípios de funcionamento, são as bases para o desenvolvimento da arquitetura proposta, a qual será implementada na secção 3.2 em CMOS. As equações da dinâmica do sistema são deduzidas para que possam ser empregadas posteriormente usando teoria de controlo. De modo a analisar o funcionamento da PLL é necessário descrever quais os blocos que a constituem, assim como o funcionamento de cada deles.

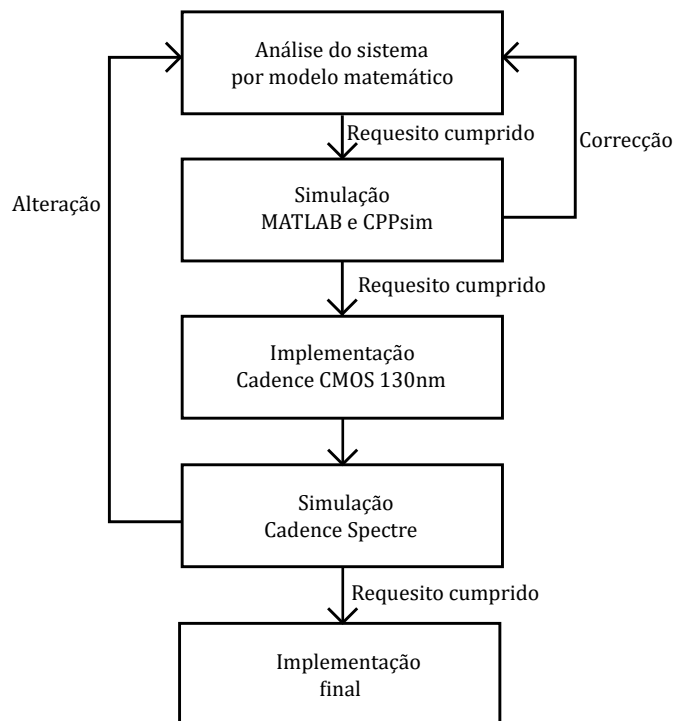


Figura 3.2: Diagrama dos passos de desenho da PLL.

O planeamento seguido para o desenho da PLL segue os passos apresentados na figura 3.2.



### 3.1 Modelo linear

O modo de operação deste circuito consiste no cálculo da diferença de fase de dois sinais  $\phi_{in}$  e  $\phi_{out}$ , sendo que devido à existência de uma realimentação negativa, a fase à saída  $\phi_{out}$  é ajustada de modo a sincronizar com a fase do sinal da entrada  $\phi_{in}$  [16].

$$\phi(t) = 2\pi \int f(t)dt \Leftrightarrow f(t) = \frac{1}{2\pi} \frac{d\phi(t)}{dt} \quad (3.1)$$

A ocorrência deste condição denomina-se *lock* e/ou *steady-state*, isto é, assumindo um estado de estabilidade, sendo que neste caso a frequência à entrada do detetor de fase é igual à da referência, entrando assim em sintonia.

$$\phi_{in} = \phi_{out} \quad (3.2)$$

$$f_{out} = N \cdot f_{ref} \quad (3.3)$$

O modo de operação é definido essencialmente em torno de uma frequência central que admite uma largura de banda correspondente a duas gamas dentro das quais a PLL consegue assegurar sintonia entre frequências, *Lock range* e *Capture range* definindo um limite para cada um, tal como apresentado na figura 3.3.

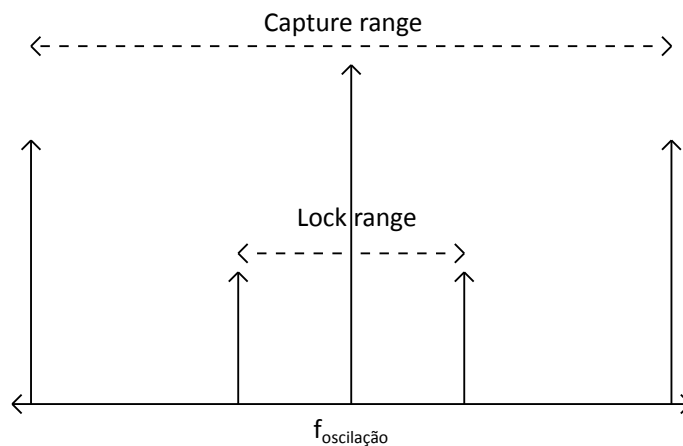


Figura 3.3: Relação do *Lock Range* e *Capture Range* em torno da frequência de oscilação.

Por *capture range*, entende-se por uma gama de frequências correspondente ao desvio de frequência do sinal entrada, sobre as quais o detetor de fase e o VCO conseguem atuar/reagir com rapidez suficiente para que seja atingido o estado de *lock* sem ter que percorrer outro ciclo de comparação da referência de entrada. Já para *lock-range* é uma gama mais curta de frequências de desvio de frequência do sinal entrada do detetor de fase, às que o oscilador retêm o valor da frequência de saída após a resposta transitória.

O sinal de frequência mais elevada  $f_{out}$  irá conservar as características desejáveis (isto é, precisão de frequência e estabilidade ao longo do tempo e temperatura) do sinal de entrada. Consequentemente, um oscilador controlado por tensão, de baixa qualidade, em conjunto com um sinal

de referência de alta qualidade, mas de menor frequência, produz uma sinal de alta qualidade e alta frequência.

### 3.1.1 Projeto da PLL a usar como caso de estudo

#### 3.1.2 Requisitos do sistema

As especificações seguintes definem o desenho da PLL assim como várias das decisões tomadas para que seja ajustado o funcionamento pretendido.

- A PLL seja capaz de sintetizar pelo menos 6 frequências numa gama entre 50-600 Hz;
- O valor para o *Jitter* de frequência à saída menor que 1%;
- Tomar atenção às limitações de área e consumo.

##### 3.1.2.1 Detetor de fase

O detetor de fase compara o sinal à saída do divisor com o sinal de referência, gerando à saída um sinal que é aproximadamente proporcional ao erro de fase  $\phi_e$ . Calcula o desvio de fase entre sinal proveniente da malha fechada com o sinal de entrada obtendo  $\Delta\phi$  em relação ao primeiro. O valor em tensão à saída é proporcional à diferença de fase entre a referência e a saída do divisor. Deste modo o funcionamento do detetor de fase pode ser equiparado a um detetor de erro. O funcionamento é então proporcional a  $\Delta\phi$ , correspondendo uma saída  $[UP, DOWN] = [1, 0]$  ao caso em que o sinal de referência está em avanço em relação ao sinal realimentado, e  $[UP, DOWN] = [0, 1]$  ao inverso, isto é, 3.4.

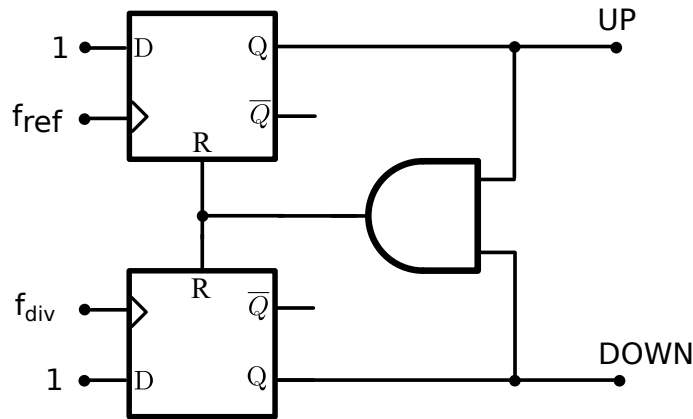


Figura 3.4: Exemplo de detetor de fase de 3 estados [16].

$$Out_{PD} \begin{cases} UP, & \Delta\phi > 0 \\ 0, & \Delta\phi = 0 \\ DOWN, & \Delta\phi < 0 \end{cases} \quad (3.4)$$

em que o erro de fase em tensão, correspondente à saída do PD, é dado por:

$$\phi_e(t) = K_P[\phi_{out}(t) - \phi_{in}(t)], \quad \text{Com } K_P \text{ o ganho do detetor de fase} \quad (3.5)$$

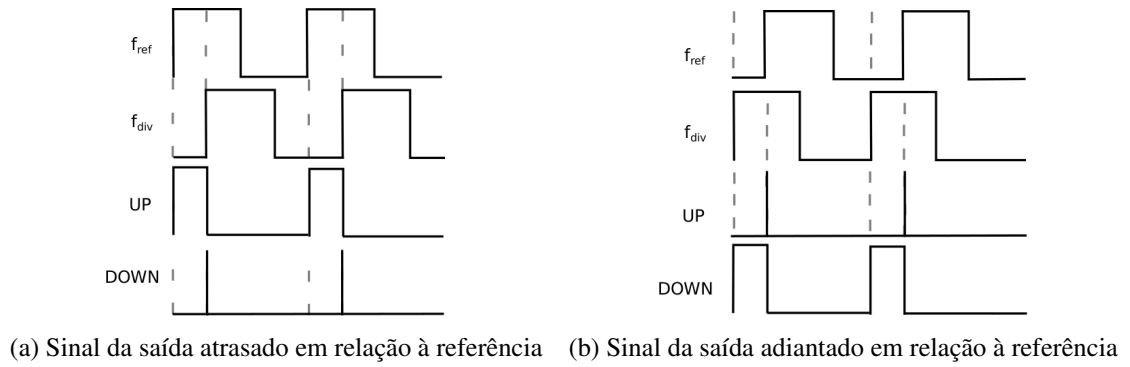


Figura 3.5: Exemplo de funcionamento do detetor de fase 3.4.

Na operação do PD, as comutações em R (*reset*) provocam um transitório (*glitch*) que pode ser interpretado por um impulso de pequena duração, que não interfere diretamente na operação, sendo que apenas a média do valor da diferença de fase é tomado com o resultado. No entanto a sua influência irá provocar erros, principalmente na tensão de controlo à entrada do oscilador [40, 41].

É este bloco que está encarregue de definir o estado da PLL, isto é, quando  $\Delta\phi = 0$  durante vários ciclos do relógio de referência é atingido o estado de *lock* [42].

### 3.1.2.2 Charge-pump

O bloco *charge pump* transforma os sinais em tensão gerados pelo detetor de fase em impulsos de corrente, consoante a variação de  $\Delta\phi$  [43].

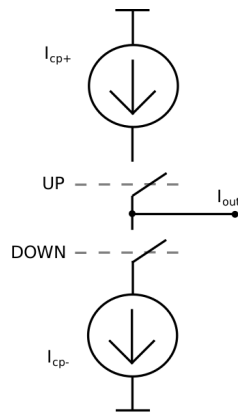


Figura 3.6: Funcionamento da *charge-pump*.

As seguintes equações modelam a resultante à saída do *charge-pump*, com a variação da fase  $\Delta\phi$  à saída do detetor de fase [43].

$$\begin{aligned} |Q| &= \int_t^{t+T_{ref}} I_{cp}(\tau) d\tau = I_{cp} \frac{\Delta\phi \cdot T_{ref}}{2\pi} \\ I_{avg} &= \frac{|Q|}{T_{ref}} = I_{cp} \frac{\Delta\phi}{2\pi} \\ K_{pd} &= \frac{I_{avg}}{\Delta\phi} = \frac{I_{cp}}{2\pi} \text{ A/rad} \end{aligned} \quad (3.6)$$

O valor médio da corrente de saída do *charge pump* é então dado por:

$$I_{out} = I_{cp} \frac{\Delta\phi}{2\pi} \quad (3.7)$$

que corresponde à seguinte tabela 3.1 dado o valor vindo do detetor de fase.

Tabela 3.1: Resultado consoante o valor de  $\Delta\phi$  para efeitos de carga no filtro.

UP	DOWN	Condição	Sentido/valor
1	0	Carrega	$I_{out}$ no sentido do filtro
0	1	Descarrega	$I_{out}$ no sentido contrário do filtro
0	0	-	$I_{out} = 0$
1	1	-	$I_{out}$ não é 0

### 3.1.2.3 Filtro

O filtro considerado é um filtro passa-baixo, podendo ser de 1ª ou 2ª ordem e ativo ou passivo.

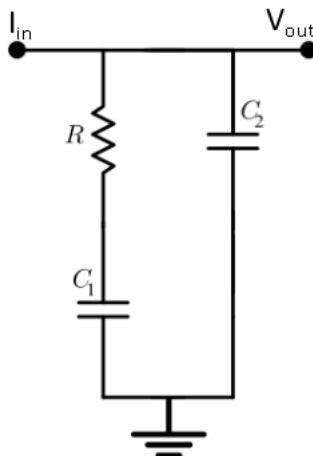


Figura 3.7: Filtro passa baixo *lead-lag* [17].

A análise dinâmica da PLL depende do comportamento deste bloco. A equação 3.8 apresenta o ganho em tensão do filtro implementado, porém como a saída do CP é em corrente temos de

usar o ganho em transimpedância 3.9.

$$\begin{aligned}
 F(s) = Z_{eq}(s) &= \left( R_1 + \frac{1}{sC_1} \right) // \frac{1}{sC_2} \\
 &= \frac{\frac{sR_1C_1+1}{s^2C_1C_2}}{\frac{sR_1C_1+1}{sC_1} + \frac{1}{sC_2}} \\
 &= \frac{sR_1C_1+1}{s(C_1+C_2)\left(\frac{sR_1C_1C_2}{C_1+C_2}+1\right)}
 \end{aligned} \tag{3.8}$$

$$Z_{eq}(s) = K_{filtro} \frac{\left(\frac{s}{\omega_z} + 1\right)}{s\left(\frac{s}{\omega_p} + 1\right)} \tag{3.9}$$

em que  $\omega_z = \frac{1}{R_1C_1}$  e  $\omega_p = \frac{C_1+C_2}{R_1C_1C_2}$ , que caso  $C_1 \gg C_2$  se pode aproximar por  $\omega_p = \frac{1}{R_1C_2}$  e o ganho vem dado por  $1/(C_1+C_2) \approx 1/C_1$  [44].

### 3.1.2.4 Oscilador controlado por tensão

O oscilador controlado por tensão (VCO) é o bloco mais importante e complexo no desenho da PLL. Este circuito oscila em torno de uma frequência central e permite varrer uma dada gama de frequências, consoante o valor de tensão de controlo à entrada.

Existem várias arquiteturas de VCO, sendo as mais comunmente usadas em PLL as realizações com osciladores LC e as baseadas em topologias de oscilador em anel — que recorrem ao uso de inversores em malha fechada como é o caso (*ring oscillator*).

A equação geral da frequência de oscilação de um oscilador do tipo LC é:

$$f_{osc} = \frac{1}{2\pi\sqrt{LC}} \tag{3.10}$$

e a de um *ring oscillator*:

$$f_{osc} = \frac{1}{\sum \text{Andares de atraso}} \tag{3.11}$$

Ambas as equações representam a frequência livre, sendo que no caso de um VCO é necessário expressar a dependência do valor de  $V_{ctrl}$ , resultante função de transferência 3.12.

$$\begin{aligned}
 \omega &= \frac{d\theta}{dt} \\
 \omega_{vco} &= \omega_{ref} \pm K_{vco}V_{in} \quad , \quad V_{in} = V_{ctrl} \\
 \theta_{out}(t) &= \omega_{vco}t \pm K_{vco} \int_{-\infty}^t v_{vco}(\tau) d\tau \\
 V_{vco_{out}} &= V \cos \left( \omega_{ref} \pm K_{vco} \int V_{vco_{in}} dt \right) \\
 H_{vco}(s) &= \frac{\omega_{vco}}{v_{vco}} = \frac{K_{vco}}{s}
 \end{aligned} \tag{3.12}$$

$$K_{vco} = \frac{\partial \omega}{\partial V} \left[ \frac{rad/s}{V} \right] \quad (3.13)$$

De modo analítico, isto é, por análise da curva de funcionamento do oscilador, é também possível expressar o valor do ganho  $K_{vco}$  através da equação 3.14.

$$K_{vco} = \left( \frac{f_{vco}(V_{ctrl_2}) - f_{vco}(V_{ctrl_1})}{V_{ctrl_2} - V_{ctrl_1}} \right) 2\pi \quad V_{ctrl_2} > V_{ctrl_1} \quad \wedge \quad V_{ctrl_i} \in Linear_{K_{vco}} \quad (3.14)$$

### 3.1.2.5 Divisor

Devido à necessidade de geração de uma gama de frequências à saída superior à da entrada, é necessário que à entrada do detetor de fase seja comparada uma frequência da mesma gama. Isto implica que a frequência à saída do oscilador seja dividida de modo a estabelecer a igualdade em 3.3. Esta divisão de frequências consiste num conjunto de atrasos, realizados por blocos que realizam uma divisão por N inteiro, ou fracionário.

A divisão inteira consiste num bloco digital, ao qual é ligado um conjunto de contadores em cadeia, operando de modo assíncrono com a frequência de entrada igual ao valor da saída anterior, acumulando assim 1 ciclo por cada ciclo de relógio, isto é caso seja feita uma divisão por 2, podendo obter-se valores de divisões superiores usando circuitos lógicos mais complexos [45, 46].

A divisão fracionaria apresenta uma arquitetura diferente, sendo geralmente realizada com moduladores do tipo  $\Sigma\Delta$  [47].

### 3.1.3 Análise dinâmica

De forma a compreender o funcionamento do sistema, é feita uma análise atendendo às leis de teoria de controlo apresentadas pelo livro em [24].

$$G(s) = \frac{K_{pd}K_{vco}F(s)}{Ns} = \frac{K_{pd}K_{vco}}{Ns} \left( \frac{sR_1C_1 + 1}{s(C_1 + C_2) \left( \frac{sR_1C_1C_2}{C_1 + C_2} + 1 \right)} \right) \quad (3.15)$$

De modo a simplificar as seguintes deduções, é tomada uma notação mais compacta para a expressão do ganho  $K_{pd}$  juntamente com o ganho do filtro e de N.

$$K_{pd_{alt}} = \frac{K_{pd}}{C_1N} = \frac{I_{cp}}{2\pi C_1N} \quad (3.16)$$

A função de transferência do sistema em malha fechada visto pela entrada do detetor de fase é dada por:

$$H(s) = \frac{\phi_{div}}{\phi_{ref}} = \frac{G(s)}{1 + G(s)} = \frac{1 + \frac{s}{\omega_z}}{1 + \frac{s}{\omega_z} + \frac{s^2}{K_{pd_{alt}}K_{vco}} + \frac{s^3}{\omega_p K_{pd_{alt}}K_{vco}}} \quad (3.17)$$

Se se considerar  $\omega_p \gg \omega_n$ , isto é, que a relação  $\frac{C_1^2}{C_1} \leq 0.1$ , a função de transferência em malha aberta é simplificada por:

$$G(s) = \frac{K_{pd_{alt}} K_{vco} F(s)}{Ns} = \frac{K_{pd_{alt}} K_{vco} \left(1 + \frac{s}{\omega_z}\right)}{s^2 \left(1 + \frac{s}{\omega_p}\right)} \quad (3.18)$$

o que resulta na função de transferência em malha fechada

$$H(s) = \frac{1 + \frac{s}{\omega_z}}{1 + \frac{s}{\omega_z} + \frac{s^2}{K_{pd_{alt}} K_{vco}}} = \frac{K_{pd_{alt}} K_{vco} \left(1 + \frac{s}{\omega_z}\right)}{K_{pd_{alt}} K_{vco} + s \frac{K_{pd_{alt}} K_{vco}}{\omega_z} + s^2} \quad (3.19)$$

Partindo da equação 3.19 é feita uma análise seguindo a aproximação a um sistema de 2ª ordem, para a função de transferência:

$$H(s)_{simples} = \frac{2\zeta \omega_n s + \omega_n^2}{\omega_n^2 + 2\zeta \omega_n s + s^2} \quad (3.20)$$

A frequência natural do sistema, também conhecida por ressonância, é dada por:

$$\omega_n = \sqrt{K_{vco} K_{pd_{alt}}} \quad (3.21)$$

e o valor do coeficiente de amortecimento (*damping factor*) é definido por:

$$\zeta = \frac{\omega_n}{2\omega_z} = \frac{R_1}{2} \sqrt{K_{vco} K_{pd_{alt}} C_1^2} \quad (3.22)$$

O tempo de assentamento (*settling time*)  $t_s$  é dado em função de  $\zeta$  e  $\omega_n$  e o erro em regime estacionário  $erro_{ss}$

$$t_s = \frac{-\ln(erro_{ss})}{\zeta \omega_n} \quad (3.23)$$

A margem de fase é dada sabendo a frequência de travessia e as frequências do polo e do zero.

$$\phi_m = \arctan\left(\frac{\omega_c}{\omega_z}\right) - \arctan\left(\frac{\omega_c}{\omega_p}\right) \quad (3.24)$$

A frequência de travessia é dada por uma aproximação da extensão do valor do zero do filtro. Este parâmetro define diretamente a banda da PLL, pelo que a condição 3.26 é geralmente verdadeira quando a banda é efetivamente reduzida em relação à frequência de referência [24].

$$\omega_c \approx \frac{\omega_n^2}{\omega_z} \quad (3.25)$$

Uma boa aproximação para saber a estabilidade em malha aberta do sistema, considerando a aproximação a um sistema linear, obtém-se comparando a frequência de travessia com a da

referência.

$$\omega_c < \frac{\omega_{ref}}{\pi(1 + \frac{\pi\omega_z}{\omega_{ref}})} \quad (3.26)$$

Caso a banda da PLL seja maior e a relação  $\frac{\omega_{ref}}{\omega_c}$  menor, é necessário fazer uma análise assumindo as leis de sistemas discretos, sendo a condição em 3.26 neste caso não aplicada.

De modo a definir os parâmetros que permitem determinar a estabilidade do sistema em regime estacionário e caso a condição 3.26 não seja regular, é necessário definir o valor de  $K'$  [17], denominado de ganho em malha aberto normalizado.

$$K' = \frac{1}{\frac{\pi\omega_z}{\omega_{ref}} \left(1 + \frac{\pi\omega_z}{\omega_{ref}}\right)} \quad (3.27)$$

Gardner [17] define os valores  $a$  e  $b$  como parâmetros usados para estabelecer limites dos valores dos polos em malha fechada, tomando uma análise a um sistema de função de transferência discreta, isto é a estabilidade restrita ao círculo unitário, ou seja,

$$\begin{aligned} b &= 1 + \frac{C_1}{C_2} \\ a &= \exp\left(-\frac{2\pi b\omega_z}{\omega_i}\right) \end{aligned} \quad (3.28)$$

em que  $b \gg 1 \Rightarrow C_1 \gg C_2$ . O critério de estabilidade de Gardner [17] mantém uma relação entre os valores da frequência de entrada  $\omega_i = 2\pi f_{ref}$ , os parâmetros  $a$  e  $b$ , e a frequência angular do zero  $\omega_z$ .

$$K' < \frac{4(1+a)}{\frac{2\pi(b-1)\omega_z}{\omega_i b} \left[ \frac{2\pi(1+a)\omega_z}{\omega_i} + \frac{2(1-a)(b-1)}{b} \right]} \quad (3.29)$$

### 3.1.4 Resultados simulação MATLAB

Os resultados apresentados foram obtidos seguindo uma metodologia teórico-experimental, em que foram feitos varrimentos de vários parâmetros dos circuitos depois de calculados os valores teóricos. Com a variação de grandezas dinâmicas na secção 3.1.3, juntamente com varrimentos de parâmetros dos circuitos, como por exemplo o  $K_{vco}$ , foi possível encontrar soluções que garantem as primeiras condições de estabilidade do sistema.

Nas simulações apresentadas a seguir foram feitas alguma assunções com base no projeto de PLL para diferentes constantes de tempo, mas que se ajustam para a PLL em causa, sendo feita uma equivalência das seguintes condições:

- Garantir que  $\omega_n \leq \omega_{ref}$
- $\omega_p \gg \omega_c \gg \omega_n \gg \omega_z$
- O valor de  $t_s$  na ordem das centenas de milissegundos dado que a gama de  $t_s \propto T_{ref}$



- O  $erro_{ss} \approx 1\%$

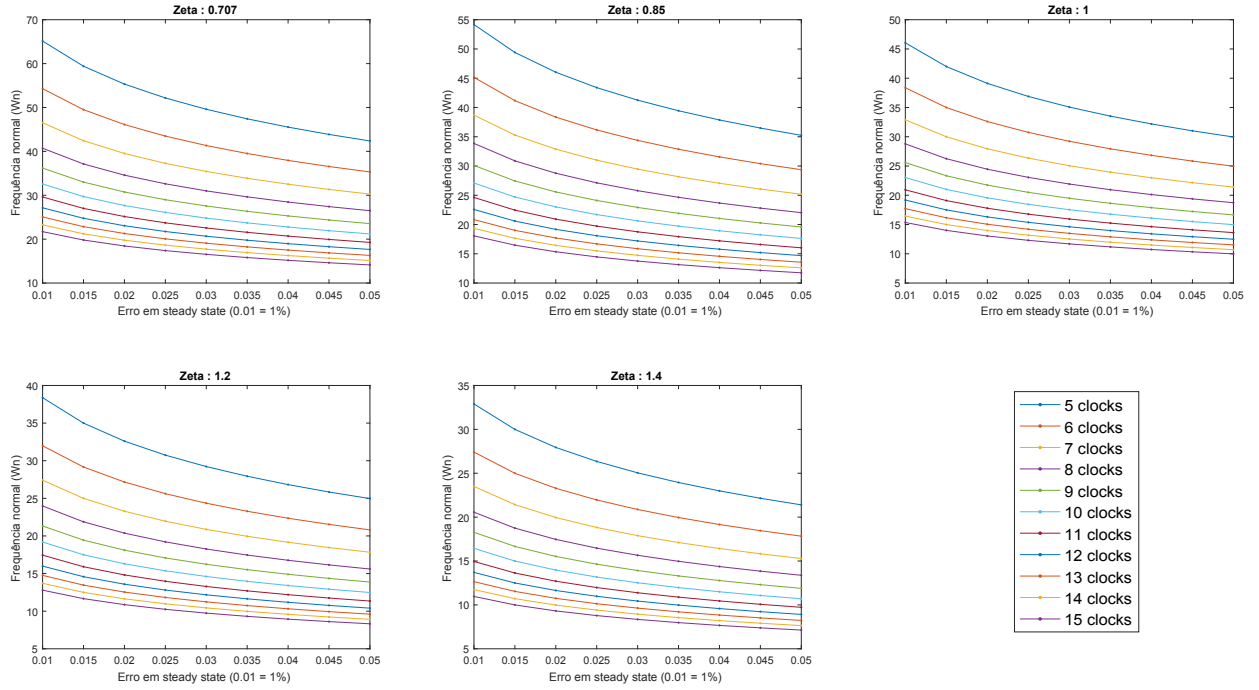


Figura 3.8: Varrimentos em função de  $N_{clocks}$ .

A cada gráfico da figura 3.8 correspondem diferentes valores de  $\zeta$ , varrendo valores do erro para os quais se considera ter sido atingido o regime estacionário  $erro_{ss}$  e o número de ciclos em relação ao período da frequência de referência  $N_{clocks}$ , como é apresentado na equação 3.30.

As curvas são respetivamente, de cima para baixo, as correspondentes ao varrimento de  $N_{clocks} = 5$  até  $N_{clocks} = 15$  com incremento de 1 relógio por passo.

$$\omega_n = \frac{-\ln(erro_{ss})}{\zeta T_{ref} N_{clocks}} \quad (3.30)$$

em que  $T_{ref} N_{clocks} = t_s$  como visto na equação 3.23.

Atendendo à função de transferência, foi feito a análise da resposta transitória, começando por assumir pontos iniciais até convergir para uma solução aceitável e passível de implementação com valores realistas, isto é, que a resultante da parametrização contenha um equilíbrio para os blocos respetivos, atendendo às condições previamente definidas no início desta subsecção.

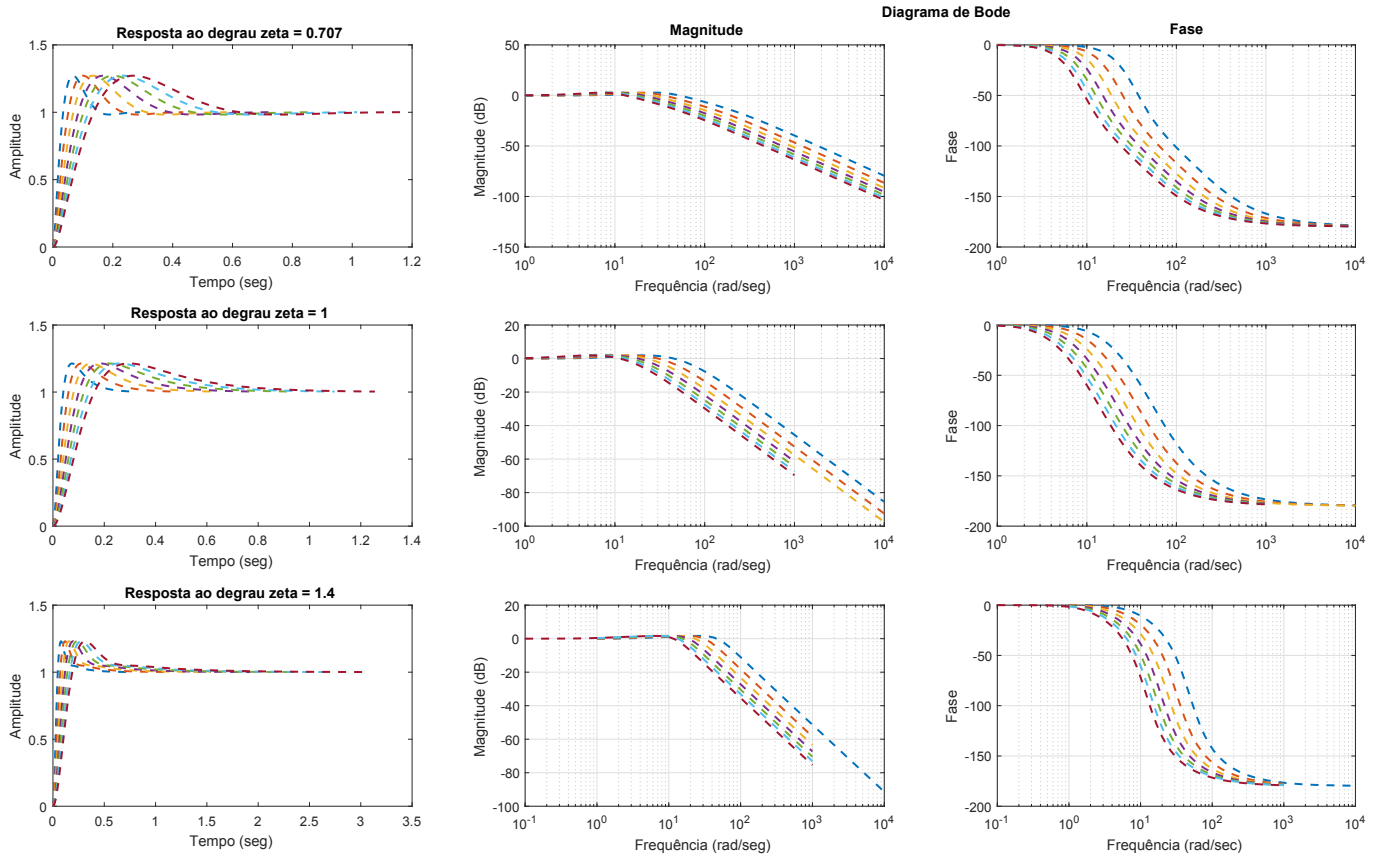


Figura 3.9: Simulações iniciais considerando  $H(s)$  um sistema de 3ª ordem.

Na simulação ilustrada em 3.9 é feito um varrimento dos valores de  $\omega_n$  da figura 3.8 com os respectivos valores de  $\zeta$ , fixando  $K_{vco} = 2\pi \cdot 10$  krad/V,  $N = 96$  e  $I_{cp} = 10 \mu A$ , sendo definidos os parâmetros do filtro 3.8 em função destes.

$$C_1 = \frac{K_{vco} I_{cp}}{2\pi N \omega_n^2}, \quad C_2 = 0,1 C_1, \quad R_1 = \frac{2\zeta}{\sqrt{\frac{K_{vco} I_{cp} C_1}{2\pi N}}} \quad (3.31)$$

Tabela 3.2: Primeiras parametrizações do filtro em relação à figura 3.9.

	$\omega_n$	10	15	20	25	30	35	40
	$C_{11} = C_{12} = C_{13} (\mu F)$	17,7	7,9	4,5	2,9	2	1,45	1,1
$\zeta_1 = 0,707$	$R_1 (k\Omega)$	8	12	16	20	24	28	32
$\zeta_2 = 1$	$R_1 (k\Omega)$	11,3	17	22,6	28,3	33,9	39,5	45,2
$\zeta_3 = 1,4$	$R_1 (k\Omega)$	16	23,7	31,6	39,5	47,4	55,3	63,3

As soluções obtidas seguindo este método, apesar de conduzirem a uma boa resposta em simulação Matlab para os diferentes valores varridos, apresentam uma banda demasiado reduzida, isto é, para desvios muito pequenos da frequência à entrada da PLL, esta pode perder a sintonia muito facilmente.

Para ultrapassar esta limitação, foi feito um ajuste diferente, isto é, ao invés de se fixar o valor de  $\zeta$ , foi fixado o valor da banda e ajustado o ganho em malha aberta, por forma a minimizar o tempo de estabelecimento.

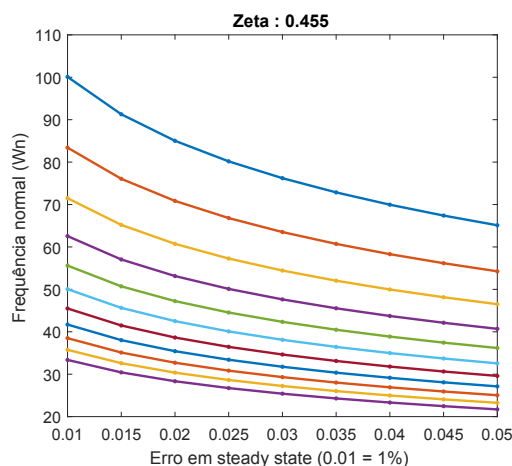


Figura 3.10: Varrimento para  $\omega_n$  superior.

Para aumentar a banda é necessário aumentar o valor  $\omega_p$  o que implica aumentar o valor de  $\omega_n$ , sendo necessário ajustar os restantes parâmetros para que esta implementação seja estável.

Na figura 3.10 é feito o mesmo varrimento para  $\omega_n$  seguido em 3.8, mas para um valor inferior de  $\zeta$ .

O resultado de  $\zeta$  apresentado em 3.10 e a resposta do sistema ilustrada nas figuras 3.11 e 3.12, foram obtidos por tentativa/erro após várias tentativas, em que o varrimento foi feito fixando a largura de banda para o valor de  $\omega_p = 2\pi \cdot 15$ ,  $K_{p_{d_{alt}}}$  e  $\omega_p = 2\pi \cdot 387,5$ , e feito um varrimento do  $K_{vco}$  de  $2\pi \cdot 15k$  até  $2\pi \cdot 70k$  com saltos de  $2\pi \cdot 5k$ .

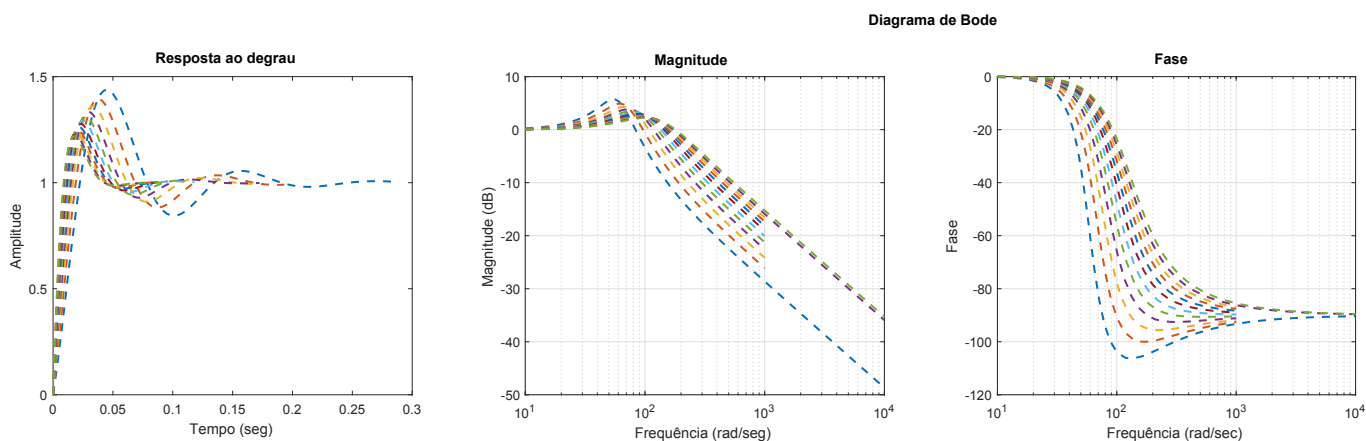


Figura 3.11: Aproximação de  $H(s)$  a um sistema de 2ª ordem.

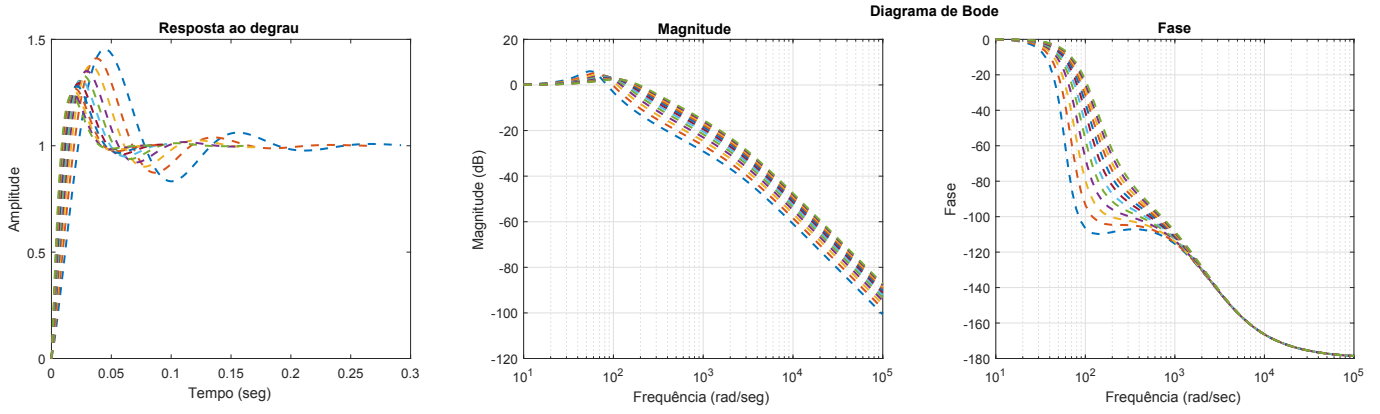


Figura 3.12:  $H(s)$  sem considerar aproximação de 2ª ordem.

A implementação final é portanto caracterizada pelos parâmetros apresentados em 3.32 e as resposta comportamentais apresentadas na figura 3.13.

$$\begin{aligned} K_{vco} &= 2\pi \cdot 65 \text{ krad/s} & N &= 96 & I_{cp} &= 10\mu A \\ R_1 &= 11,789\text{k}\Omega & C_1 &= 900\text{nF} & C_2 &= 35\text{nF} \end{aligned} \quad (3.32)$$

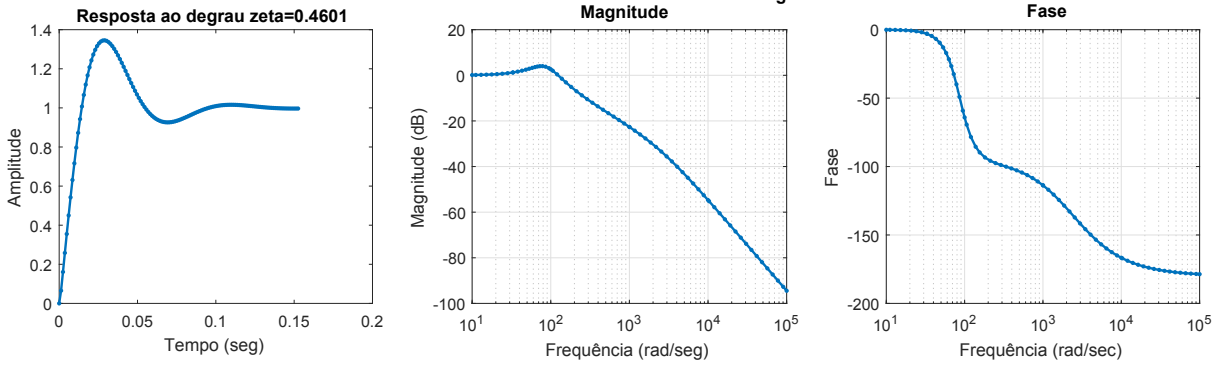


Figura 3.13:  $H(s)$  implementado.

Para caracterizar a análise dinâmica da PLL, os valores de 3.32 são substituído nas equações da secção 3.1.3.

$$\begin{aligned} \omega_n &= 86,763 \text{ rad/s} & \omega_c &\approx 104 \text{ rad/s} & \zeta &\approx 0.4601 \\ \omega_p &= 2\pi \cdot 387,5 \text{ rad/s} & \omega_z &= 2\pi \cdot 15 \text{ rad/s} & \phi_m &\approx 45^\circ \end{aligned} \quad (3.33)$$

Quanto à estabilidade apresentada pela equação 3.29 é obtido que:

$$\frac{4(1+a)}{\frac{2\pi(b-1)\omega_z}{\omega_i b} \left[ \frac{2\pi(1+a)\omega_z}{\omega_i} + \frac{2(1-a)(b-1)}{b} \right]} = 0.57862$$

$$K' = 0.34663$$

$$0.34663 < 0.57862 \quad c.q.d$$
(3.34)

### 3.1.5 Resultados de simulação em CppSim

O CppSim<sup>®</sup> é simulador comportamental, criado por Michael Perrott [15], permite a simulação de blocos complexos descritos pela sua função de transferência (figura 3.14a) expressa em linguagem orientada a objetos C++.

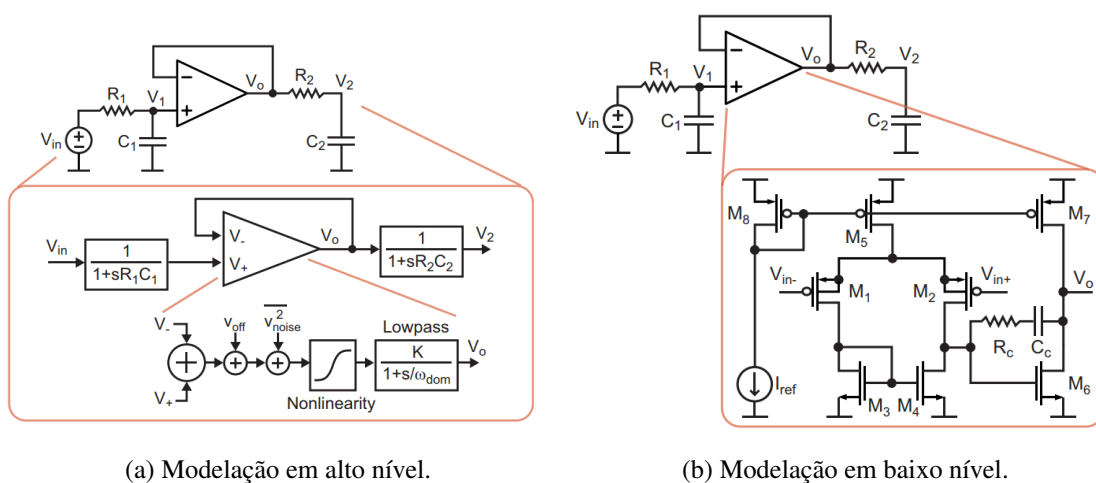


Figura 3.14: Processo de modelação dos blocos em CppSim<sup>®</sup> [15].

Para modelações mais detalhadas é possível incluir ficheiros SPICE e Verilog<sup>®</sup>, por forma a pormenorizar os parâmetros de definição do circuito em causa e das interconexões de elementos básicos, como é o caso ilustrado na figura 3.14b. O uso deste simulador, durante o trabalho realizado, facilitou a implementação e simulação de sistemas complexos, de modo a obter um conjunto-solução inicial de boa qualidade para caracterização do comportamento da PLL.

As simulações realizadas com este simulador foram feitas de modo a verificar os valores dos parâmetros obtidos em Matlab com um modelo mais próximo da implementação final.

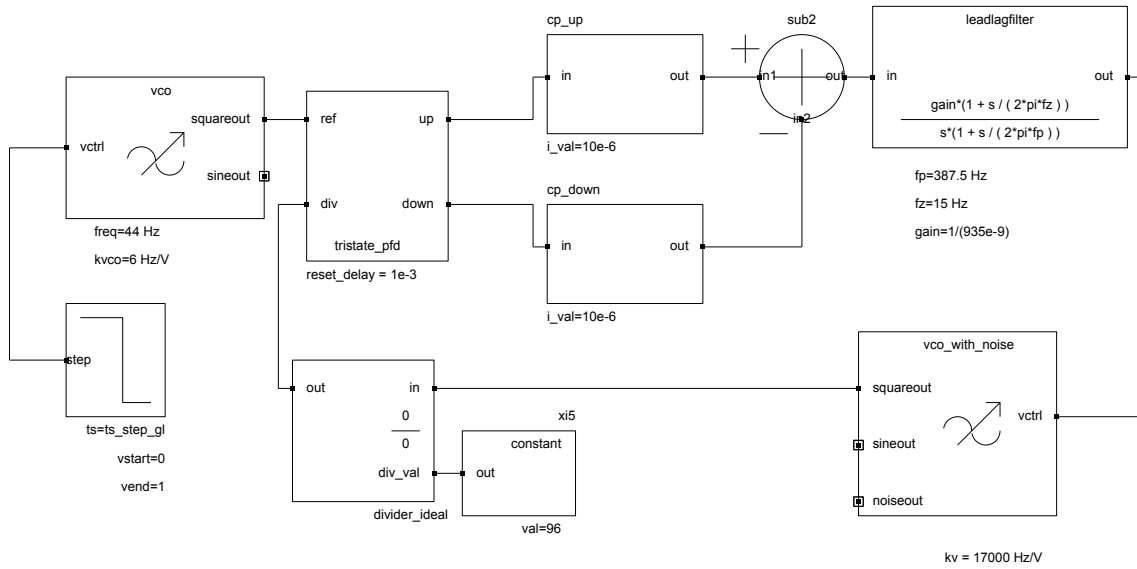


Figura 3.15: Esquemático do modelo CppSim da PLL[15].

O aumento da largura de banda da PLL para valores para os quais  $banda_{pll} > \omega_{ref}/10$ , como referido em 3.1.3 pelo critério de estabilidade de Gardner [17], requer uma atenção especial no que diz respeito à estabilidade, uma vez que a maioria dos simuladores faz uma aproximação ao modelo de 2ª ordem, não tendo em conta a transformada discreta como é feito na equação 3.34. Esta (possível) consideração, na implementação em causa, leva ao aparecimento de resultados inconclusivos para implementações em limites mais rigorosos de estabilidade, sendo só possível obter uma solução mais próxima da real com um simulador mais rigoroso, os quais podem ser comparados com os resultados da implementação final no simulador Cadence Spectre®, tal como será descrito na secção 3.2.6. No entanto, para simulação de variações da frequência de entrada, o simulador CppSim apresenta resultados satisfatórios, que permitem tirar conclusões para uma primeira implementação em CMOS do circuito simulado em alto nível.

Na figura 3.16 é feito a título de exemplo uma realização da tabela 3.2, tomando:

$$\begin{aligned}
 C_1 &= 1,45 \mu\text{F} & C_2 &= 0,1C_1 & R_1 &= 55,3 \text{ k}\Omega & (3.35) \\
 \Rightarrow \omega_c &= 2\pi \cdot 19.8 \text{ rad/s} & \omega_p &= 2\pi \cdot 1.98 \text{ rad/s} \\
 K_{vco} &= 10 \text{ kHz/V} & I_{cp} &= 10 \mu\text{A} & N &= 96
 \end{aligned}$$

As respostas apresentadas nas figuras 3.16 e 3.17 mostram que uma PLL com maior banda adapta-se melhor a pequenos desvios da entrada, sem perder a sintonia, apresentando apenas um erro que é diretamente proporcional ao desvio da frequência à entrada. É também possível verificar na figura 3.16 que o *overshoot* é também maior.

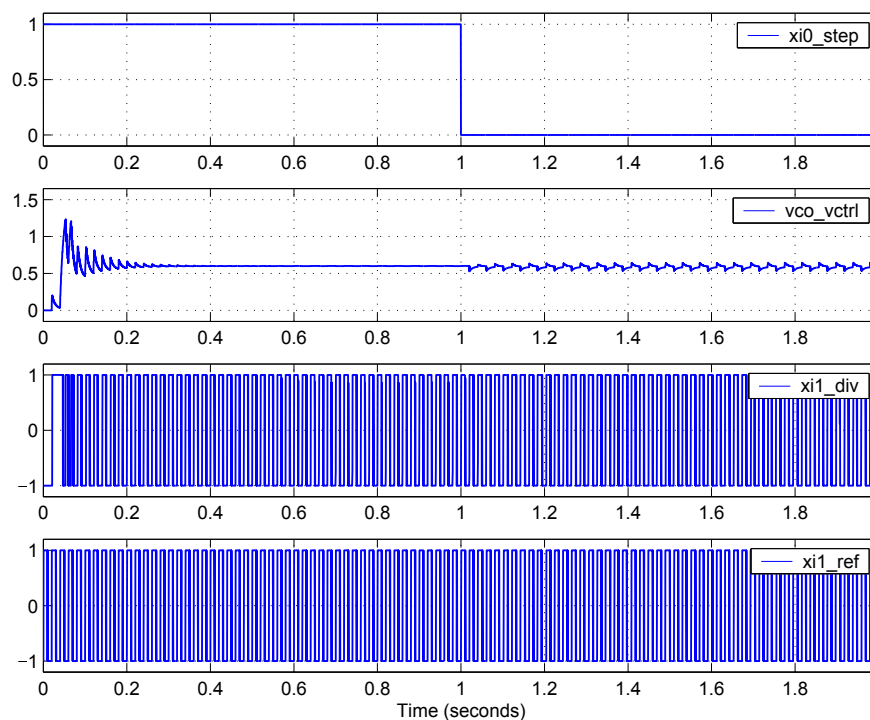


Figura 3.16: Resposta a um degrau de 1 Hz em  $F_{ref}$  para os resultados em 3.35.

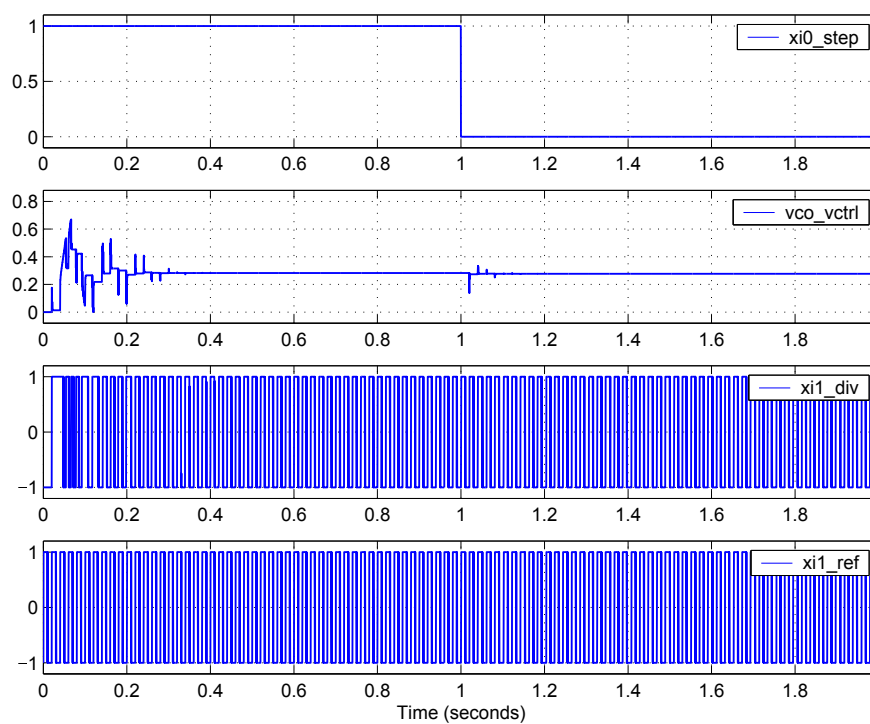


Figura 3.17: Resposta a um degrau de 1 Hz em  $F_{ref}$  — resultados em 3.32 com a exceção de  $K_{vco} = 17$  kHz.

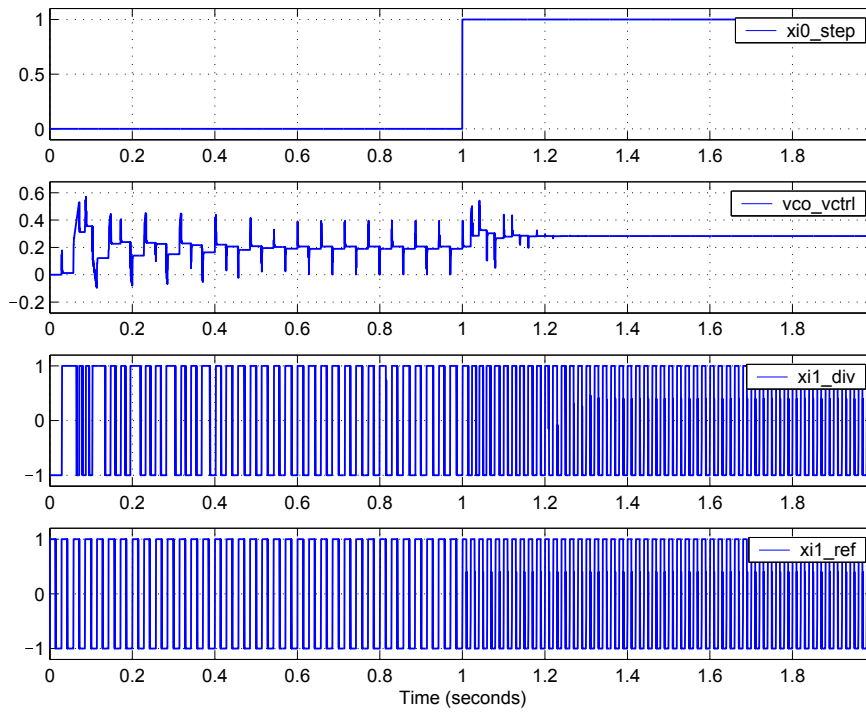


Figura 3.18: Resposta ao degrau para  $K_{vco} = 17$  kHz.

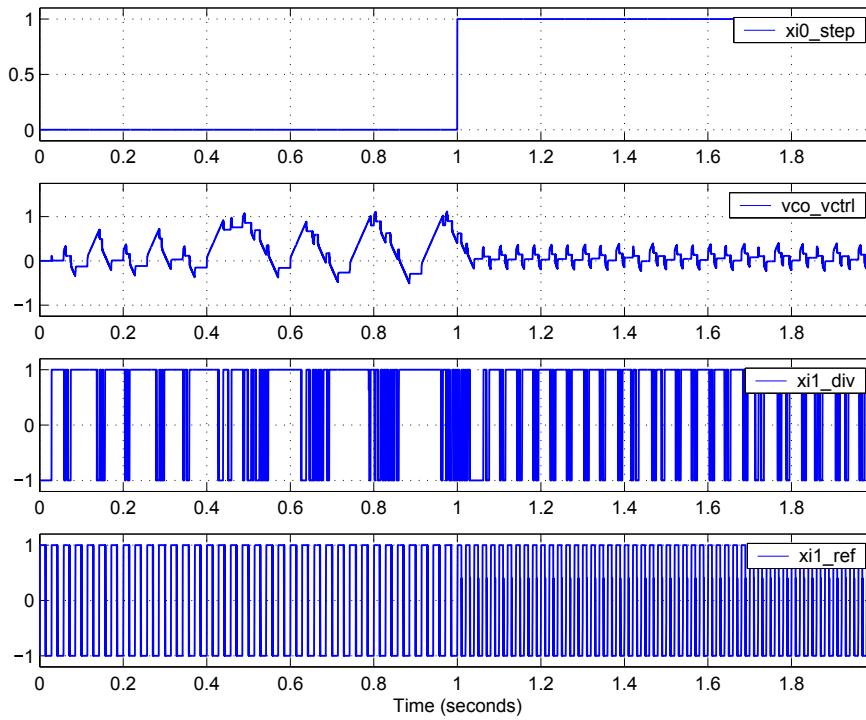


Figura 3.19: Resposta ao degrau para  $K_{vco} = 65$  kHz para a parametrização final em 3.32.



Algumas realizações da tabela são bastante estáveis e até apresentam uma banda melhor que a da figura 3.16, no entanto apresentam um tempo de estabelecimento muito superior como é o seguinte:

$$\begin{aligned}
 C_1 &= 17,7 \mu\text{F} & C_2 &= 0,1 C_1 & R_1 &= 16,3 \text{ k}\Omega & (3.36) \\
 \Rightarrow \omega_z &= 2\pi \cdot 0,5557 \text{ rad/s} & \omega_p &= 2\pi \cdot 5,557 \text{ rad/s} \\
 K_{vco} &= 10 \text{ kHz/V} & I_{cp} &= 10 \mu\text{A} & N &= 96
 \end{aligned}$$

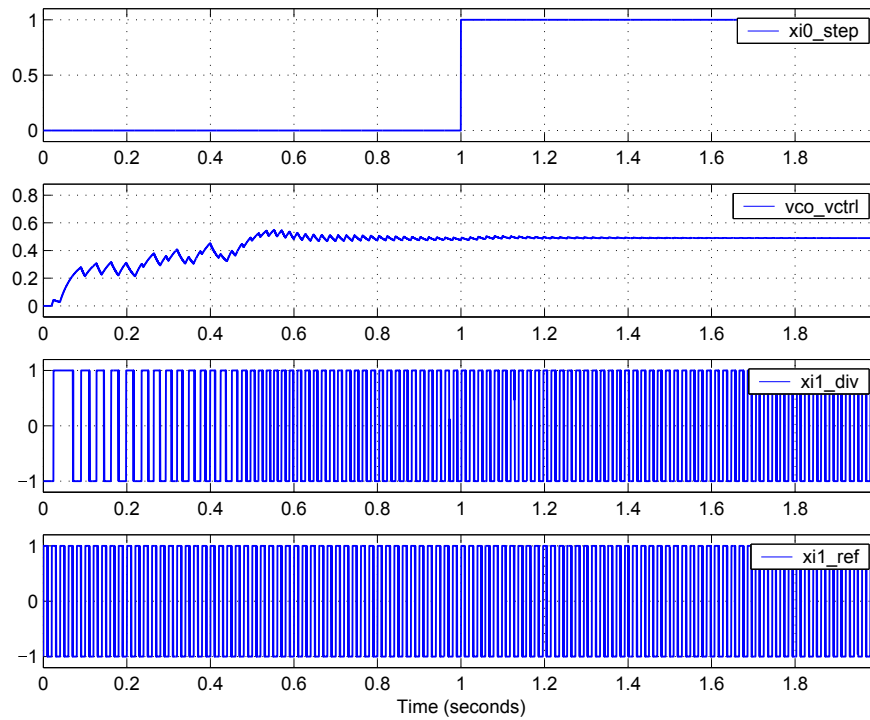


Figura 3.20: Reposta a um degrau de 1 Hz em  $F_{ref}$  para os valores de 3.36

O CppSim apresenta uma boa caracterização da resposta do sistema ao degrau, no entanto para valores de  $K_{vco}$  muito superiores (Fig. 3.19), em comparação com a frequência que deve atingir à saída após transitório, entra em condições em que  $V_{ctrl}$  atinge valores negativos, perdendo (por vezes) assim a convergência desejada.

## 3.2 Implementação dos blocos

A implementação do circuito foi feita em ambiente Cadence Virtuoso<sup>®</sup> usando uma tecnologia de 130 nm para uma tensão de alimentação de  $V_{dd} = 1,2\text{ V}$ .

### 3.2.1 Detetor de fase

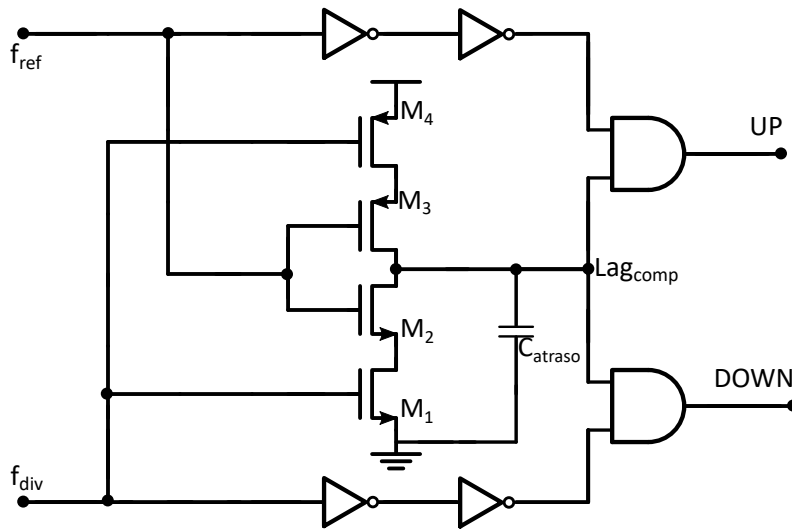


Figura 3.21: Detetor de fase [18].

O detetor de fase implementado é apresentado na figura 3.21. O valor lógico no nó  $Lag_{comp}$  é o complemento "atrasado" da comparação das entradas  $f_{ref}$  e  $f_{div}$ , isto é, os sinais são atrasados durante a comparação por uma capacidade  $C_{atraso} = 50\text{ pF}$  e por uma cadeia de inversores, que operam como *buffers* de sincronização. As saídas UP e DOWN são geradas seguindo o princípio apresentado em 3.1.2.1.

Tabela 3.3: Resultado em  $Lag_{comp}$  pelas entradas  $f_{ref}$  e  $f_{div}$ .

$f_{ref}$	$f_{div}$	$Lag_{comp}$
1	1	0
0	0	1
1	0	Mantém
0	1	Mantém

Tabela 3.4: Dimensionamento do detetor de fase.

Transístor	Tipo	$W(\mu m)$	$L(nm)$
$M_1 M_2$	NMOS	0,32	120
$M_3 M_4$	PMOS	1,52	120

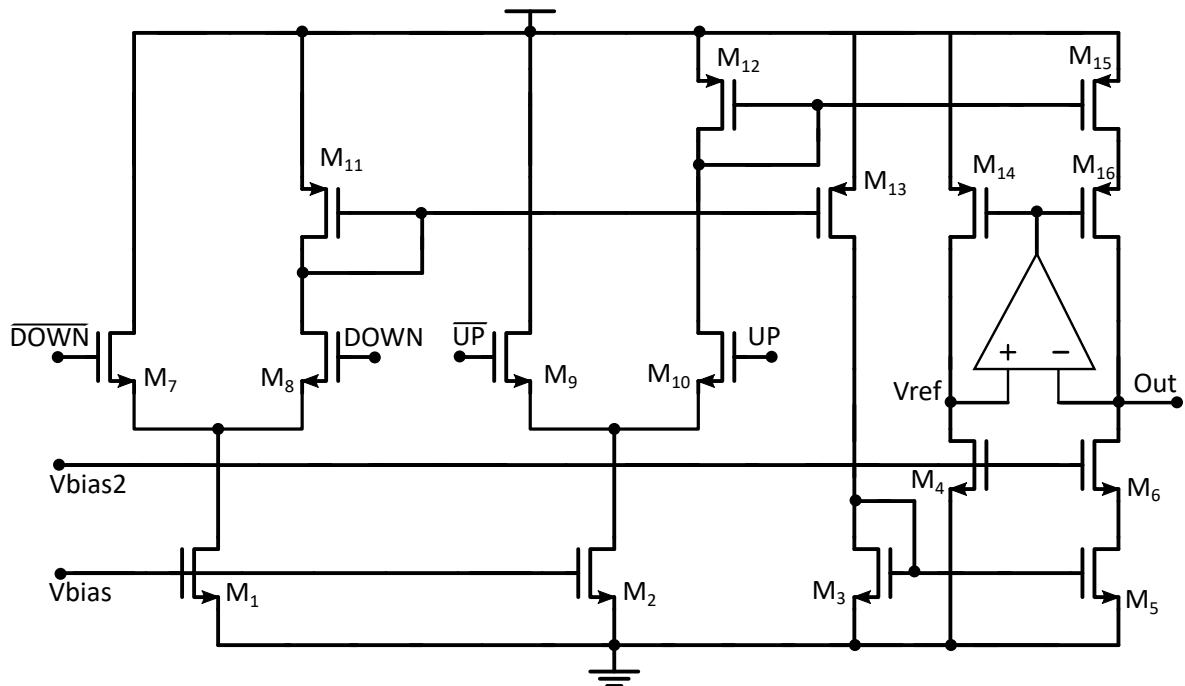
O dimensionamento das portas lógicas é feito para valores mínimos das dimensões da tecnologia (L). Em todas as portas lógicas da PLL segue-se a parametrização apresentada em 3.5, exceto no caso de transístores em série, para os quais é ajustado o valor de W por forma a manter a relação  $\beta = \frac{W_{pmos}}{W_{nmos}} = 4.75$ , esta que é resultante da comutação de 1 para 0 (e vice-versa) de um inversor simples em CMOS, para o qual o valor da comutação seja  $V_{dd}/2$  [45].

Tabela 3.5: Dimensionamento das portas lógicas da PLL.

Transístor	Tipo	W( $\mu m$ )	L(nm)
Todos	NMOS	0,16	120
Todos	PMOS	0,76	120

### 3.2.2 Charge-pump de duplo andar

O bloco da charge-pump segue uma arquitetura diferencial puramente CMOS, geralmente usada para se obter maior exatidão a baixas frequências, devido à menor sensibilidade a efeitos não ideais como *current mismatch*, modulação do canal, *clock feedthrough* e injeção de carga [16] [19].

Figura 3.22: Circuito do *charge-pump* [19] .

O modo de funcionamento 3.23 consiste na comutação alternada entre dois andares, sendo a corrente gerada pelos transístores  $M_1$  e  $M_2$  dos primeiros andares diferenciais espelhada para o andar de saída.

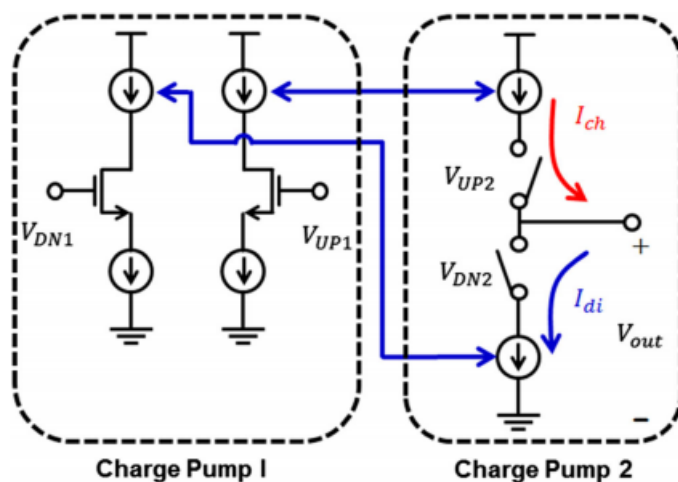


Figura 3.23: Funcionamento *charge-pump* [19].

Para diminuir os efeitos não ideais, como é o caso do desemparelhamento das correntes de descarga e de carga, que podem aparecer como ruído e *spurs* de referência no espectro do sinal à saída, é proposto o uso de um circuito de compensação juntamente com o 2º andar. Este circuito é composto por um amplificador operacional cujas entradas estão ligadas entre  $V_{ref}$  e  $V_{out}$ , atuando como um seguidor de tensão entre estas. Ao manter um nível de emparelhamento entre as tensões destes dois pontos é feita uma regulação da corrente à saída, diminuindo assim possíveis interferências à entrada do VCO.

Tabela 3.6: Dimensionamento do *charge-pump*.

Transístor										Tipo	W( $\mu m$ )	L(nm)
$M_1$	$M_2$	$M_3$	$M_4$	$M_5$	$M_6$	$M_7$	$M_8$	$M_9$	$M_{10}$	NMOS	0.16	120
$M_{11}$	$M_{12}$	$M_{13}$	$M_{14}$	$M_{15}$	$M_{16}$					PMOS	0,76	120
$M_{15}$	$M_{16}$									PMOS	1,1	120

As tensões  $V_{bias} = 450$  mV e  $V_{bias2} = 700$  mV permitem ajustar o ponto de funcionamento dos transístores que geram as correntes de polarização.

Relativamente ao desenho do amplificador operacional, foi desenhado uma arquitetura simples, sendo que o requerimento apresentado por [19] sugere que o ganho 3.37 seja o suficientemente elevado, para que as pequenas diferenças entre os nós  $V_{ref}$  e  $Out$  sejam amplificadas.

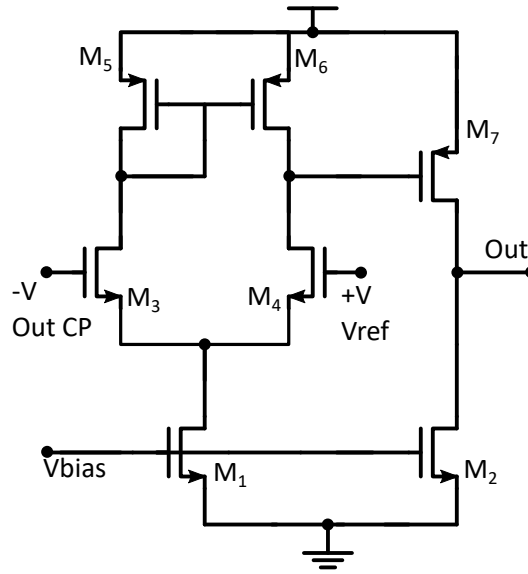


Figura 3.24: Amplificador operacional.

O ganho em tensão, saída não diferencial / entrada diferencial, para  $M_5 = M_6$  e  $M_3 = M_4$ , é.

$$\begin{aligned}
 A_{OTA} &= \left( \left( -\frac{g_{m3}}{2} - \frac{g_{m6}}{2} \right) (r_{o4} // r_{o6}) \right) \cdot (-g_{m7} (r_{o7} // r_{o2})) \\
 &= g_{m_{3,6}} g_{m7} (r_{o4} // r_{o6}) (r_{o7} // r_{o2})
 \end{aligned} \tag{3.37}$$

Tabela 3.7: Dimensionamento do amplificador operacional.

Transístor	Tipo	W( $\mu m$ )	L(nm)
$M_7$	PMOS	15,2	120
$M_5 \ M_6$	PMOS	0,76	120
$M_3 \ M_4$	NMOS	0,16	120
$M_1$	NMOS	0,32	120
$M_2$	NMOS	3,2	120

Por forma a avaliar a solução apresentada foi feito uma simulação em que é forçada a injeção e retirada de carga no filtro, de modo a observar o valor da corrente à saída para diferentes valores de  $V_{ctrl}$ .

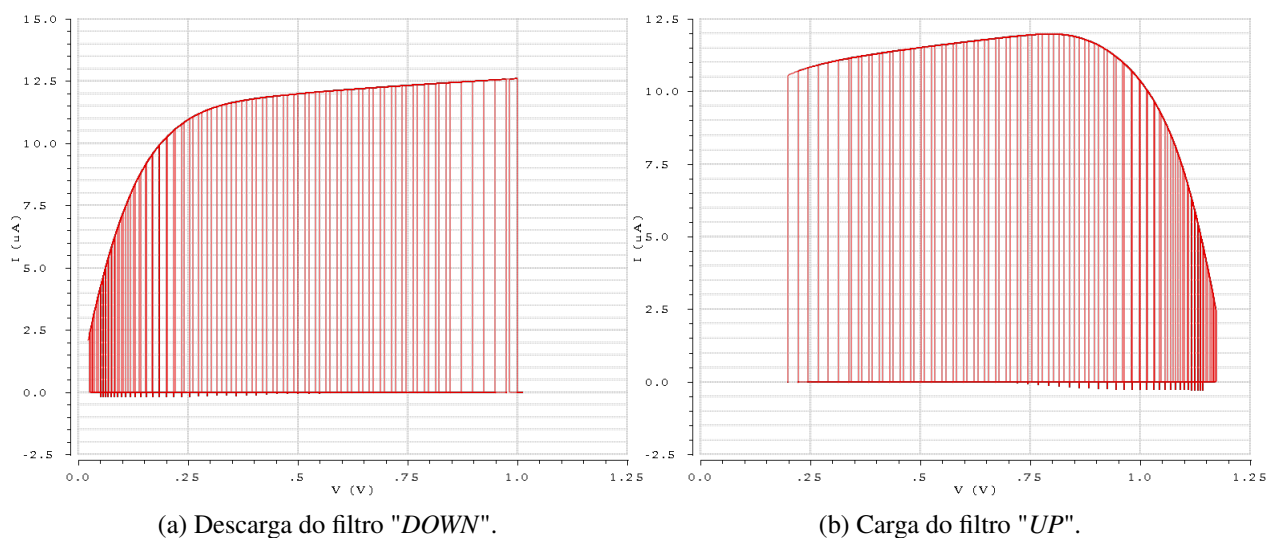


Figura 3.25: Caracterização da corrente  $I_{cp}$  do *charge-pump*.

Como se pode verificar na figura 3.25, a corrente apresenta uma curva mais plana para os níveis de tensão (média) pretendidos para o funcionamento do filtro.

O declive para valores menores que 0,2 V em 3.25a e para valores superiores a 1 V em 3.25b, devem-se a limites de funcionamento do circuito 3.22 assim como também a efeitos da não linearidade da carga/descarga da capacidade  $C_1$ .

O valor da corrente à saída do *charge-pump* está muito próximo do desenhado, apresentando um valor de  $I_{cp} \approx 11,5 \mu A$  correspondendo a uma variação de 9% com um desemparelhamento de 5%, para ambas as situações *DOWN* em 3.25a e *UP* em 3.25b, considerando a zona de interesse de funcionamento entre 0,2 V e 1 V.

### 3.2.3 Filtro *lead-lag*

O filtro implementado corresponde ao da figura 3.26, ao qual corresponde a função de transferência apresentada na secção 3.1.3, para  $C_1 = 900 \text{ nF}$ ,  $C_2 = 35 \text{ nF}$  e  $R_1 = 11,789 \text{ k}\Omega$ .

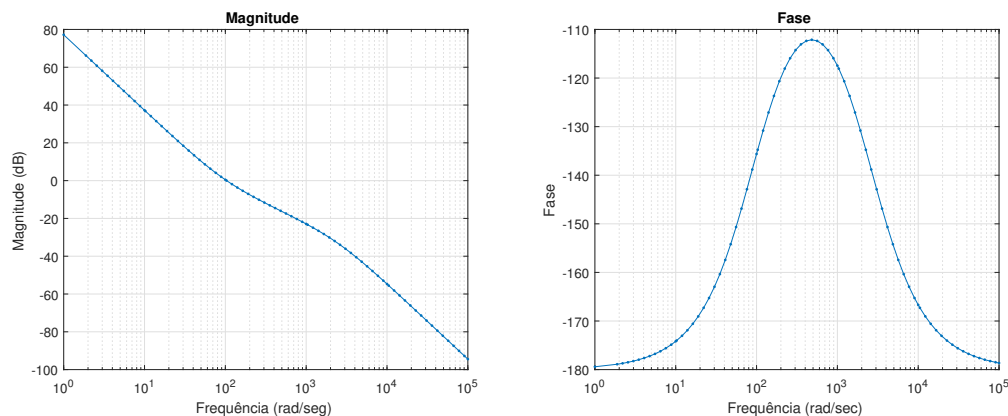
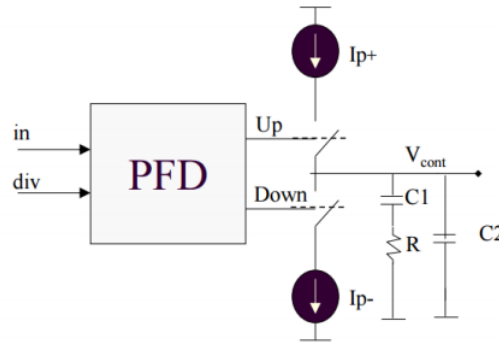
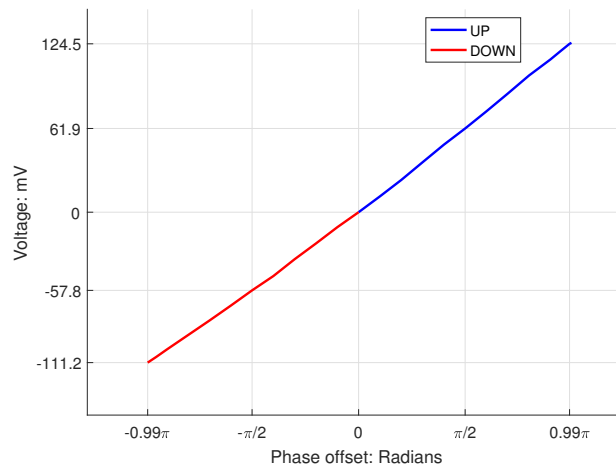


Figura 3.26: Diagrama de Bode para o filtro.

Por forma a caracterizar o conjunto detetor de fase, *charge-pump*, e filtro foi feito um varrimento do erro de fase  $\Delta\phi$ , de 0 e  $\pi$  (Fig. 3.27a). É possível verificar algumas características que anteriormente seria difícil de obter, como é o caso desta implementação não apresentar uma zona morta para o detetor de fase e a limitação de  $\pi$  [18].



(a) Figura de um detetor de fase, *charge pump* e filtro passa baixo adaptada de [48].



(b) Ganho do detetor de fase, *charge-pump* e filtro.

Figura 3.27: Caracterização da resultante do ganho  $K_{pd}$  e do ganho do filtro.

### 3.2.4 Oscilador em anel

A arquitetura do oscilador em anel é comumente usada em PLL, dada a simplicidade de projeto, apresentar consumo reduzido, e, para a frequência em questão, evitar o uso de indutâncias diminuindo a área total de implementação. Este é o bloco que tende a apresentar a maior sensibilidade a desvios PVT, dado incluir na sua constituição o maior número de elementos analógicos presentes numa PLL [49, 50],

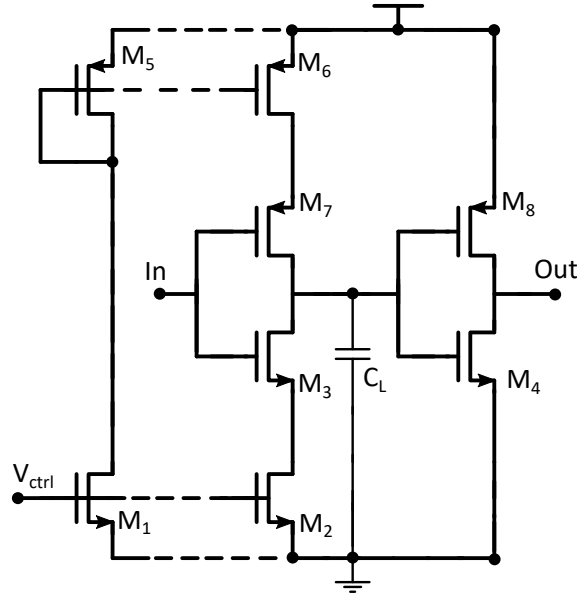


Figura 3.28: Circuito da célula de atraso e a respetiva fonte de corrente.

A célula de atraso é composta por 2 inversores e uma capacidade  $C_L = 10 \text{ pF}$  — o andar *current starved* composto pelos transístores  $M_2$ ,  $M_3$ ,  $M_6$ , e  $M_7$ , e um inversor convencional ( $M_8$  e  $M_9$ ). Quando o valor de  $In = 0$  os transístores em condução do andar *current starved* são o  $M_6$  e  $M_7$ , carregando a capacidade  $C_L$ , esta que é descarregada quando  $In = 1$ .

O transístor  $M_1$  funciona como fonte de corrente, esta que é espelhada para o transístor  $M_2$ , o mesmo acontece para  $M_5$  e  $M_6$ .

$$\begin{aligned} In = 0 & \quad , \quad V_{sd6} > V_{sg6} - V_{th6} \\ In = 1 & \quad , \quad V_{ds2} > V_{gs2} - V_{th2} \end{aligned} \quad (3.38)$$

Tabela 3.8: Dimensões da célula de atraso.

Transístor	Tipo	W( $\mu m$ )	L( $\mu m$ )
$M_1$ $M_2$	NMOS	0,16	10
$M_3$	NMOS	0,16	1
$M_4$	NMOS	0,32	0,12
$M_5$ $M_6$	PMOS	0,76	10
$M_7$	PMOS	0,76	1
$M_8$	PMOS	1,52	0,12

Cada célula apresenta um atraso dado pela soma dos atrasos de cada andar,  $t_{effect} = t_{delay_{cs}} +$



$t_{delay_{inv}}$ , em que

$$\begin{aligned} t_{delay_{cs}} &\propto \frac{2C_L}{\mu_{n2,6} C_{ox2,6} (W_{2,6}/L_{2,6}) (V_{gs2,6} - V_{th2,6})^2} \\ &\propto \frac{C_L}{i_{2,6}} \end{aligned} \quad (3.39)$$

A capacidade  $C_{L_{inv}}$ , vista pelo terminal *Out* é dada pelas capacidades intrínsecas dos transístores  $M_4$  e  $M_8$ . Aplicando o teorema de Miller resulta  $C_{L_{inv}} \approx 2(C_{gd4} + C_{gd8})$ . Para o inversor simples tem-se

$$\begin{aligned} t_{delay_{inv}} &\propto \frac{2C_{L_{inv}}}{\mu_{n4,8} C_{ox4,8} (W_{4,8}/L_{4,8}) (V_{DD} - V_{th4,8})^2} \\ &\propto \frac{2(C_{gd4} + C_{gd8})}{i_{4,8}} \end{aligned} \quad (3.40)$$

A corrente  $i_{4,8}$  é resultante de uma comutação rápida, não sendo limitada por nenhum transístor, como o é no andar *current starved*, de que resulta um pico de curta duração.

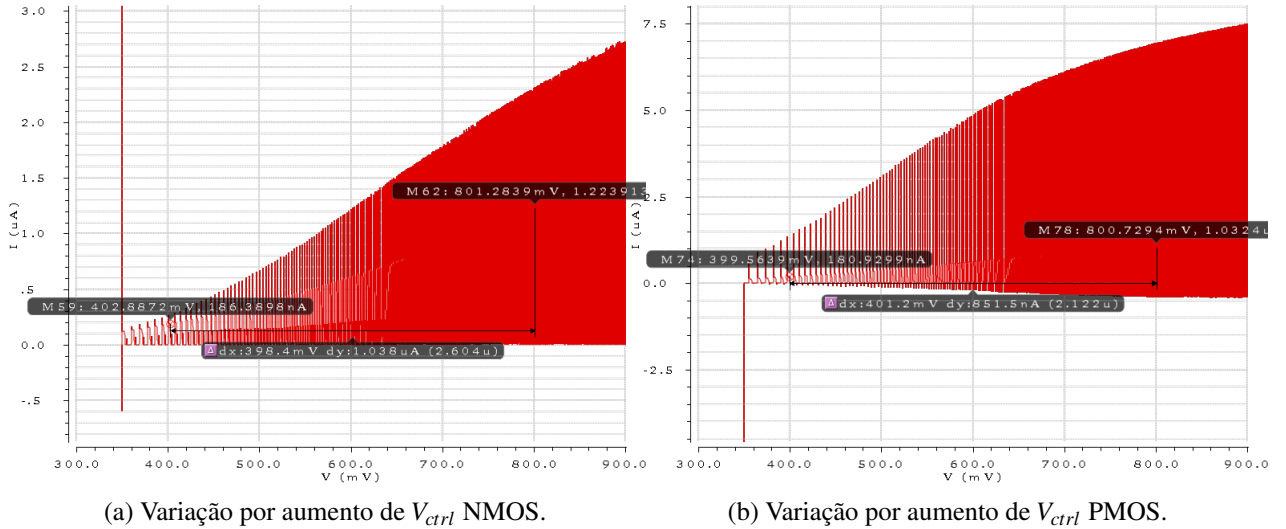


Figura 3.29: Corrente aos terminais do andar *current starved*.

Nas figuras 3.29a e 3.29b são ilustrados os valores da corrente no dreno dos transístores  $M_2$  e  $M_7$  em função de  $V_{gs1} = V_{ctrl}$ , os picos da corrente são resultantes da transição de 1 para 0 e vice-versa.

Como  $C_L \gg C_{L_{inv}}$  e  $i_{4,8} \gg i_{2,6}$ , o andar *current starved* é o que efetivamente contribui para o atraso total do célula e por consequência do oscilador, sendo então  $t_{effect} = t_{delay_{cs}}$ . Apesar de não ter um efeito de acréscimo no  $t_{effect}$ , o andar de inversores simples é necessário para o correto funcionamento do circuito, atuando como *buffers* de corrente, impedindo que transitórios de corrente que circulam nas capacidades  $C_{gd}$  dos andares *current starved* a jusante gerem transitórios de tensão nas capacidades anteriores.

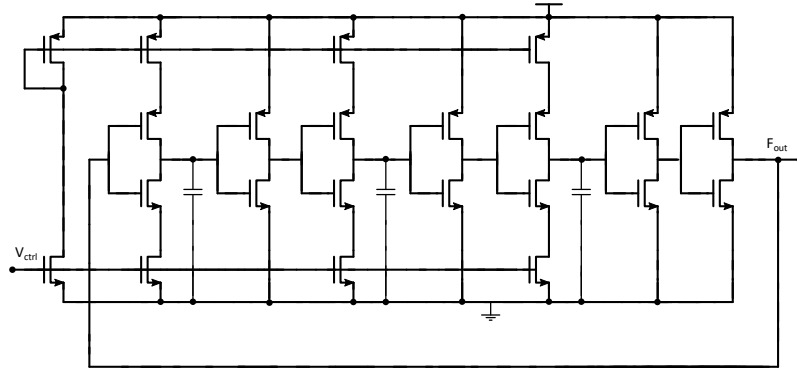


Figura 3.30: Esquemático VCO [5].

O bloco implementado contém 3 células de atraso, e um último inversor à saída, para que esta seja formatada em amplitude. Sendo portanto composto por  $N_{cell} = 3$  inversores *current starved*, 4 inversores casuais e 3 capacidades  $C_L$ . A equação de funcionamento do oscilador controlado por tensão em 3.30 é portanto uma aproximação com um erro muito pequeno, proporcional às capacidades intrínsecas dos transístores.

$$\begin{aligned}
 f_{vco}(V_{ctrl}) &= \frac{1}{2N_{cell} \cdot t_{effect}(V_{ctrl})} \\
 &\approx \frac{i_1(V_{ctrl})}{2 \cdot 3C_L} \\
 &\approx \frac{\mu_{n1} C_{ox1} (W_1/L_1) (V_{ctrl} - V_{th1})^2}{2} \cdot \frac{1}{2 \cdot 3C_L}
 \end{aligned} \tag{3.41}$$

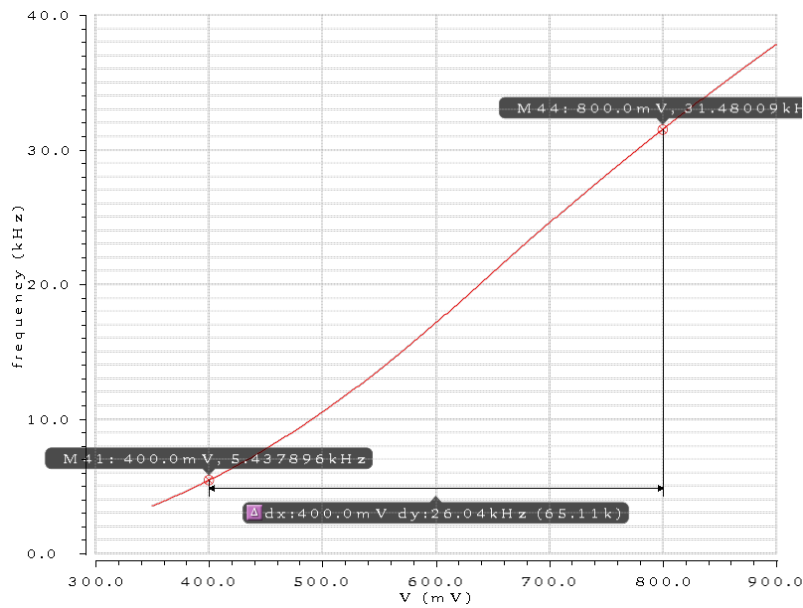


Figura 3.31:  $K_{vco}$  do oscilador.

O ganho  $K_{vco}$  do oscilador é apresentado na figura 3.31 em que :

$$K_{vco} = \frac{f_{vco}(0.8) - f_{vco}(0.4)}{0.8 - 0.4} \quad (3.42)$$

$$\approx 65 \text{ kHz/V}$$

### 3.2.5 Topologia do divisor-N inteiro

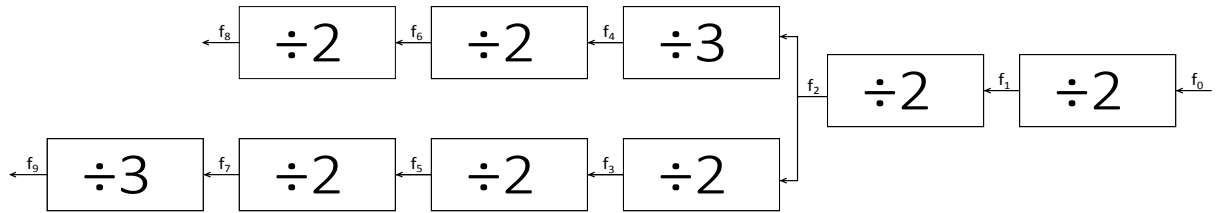


Figura 3.32: Circuito da malha N de divisores.

O bloco N da PLL consiste numa cadeia de divisores  $\div 2$  e  $\div 3$ , operando como um contador assíncrono de  $N = 96$  do sinal de entrada, isto é  $f_{vco} = 96 \cdot f_{ref}$ . Dadas as especificações de geração de frequência estabelecidas, são necessários dois canais de divisores de modo a obter uma maior resolução de frequências à saída. Um dos canais é ligado à malha de realimentação, e outro aproveita uma saída de um bloco de modo a reaproveitar a frequência em  $f_2$ . Para além do uso de divisores do tipo  $\div 2$  é também necessário o uso de divisores  $\div 3$ , visto que é necessário usar divisões diferentes de 2 de modo a obter diferentes valores para as frequências menores.

Em regime estacionário, as frequências dos sinais representados nas figuras 3.32 e 3.34 correspondem aos valores dados na tabela seguinte e na figura 3.33, respetivamente. De modo a comparar a sincronização, o último sinal da figura 3.33 corresponde ao sinal da frequência de referência e o penúltimo ao sinal  $f_9$ .

Tabela 3.9: Frequências à saída do bloco de divisor N.

Saída	Frequência (Hz)
$f_0$	4800
$f_1$	2400
$f_2$	1200
$f_3$	600
$f_4$	400
$f_5$	300
$f_6$	200
$f_7$	150
$f_8$	100
$f_9$	50

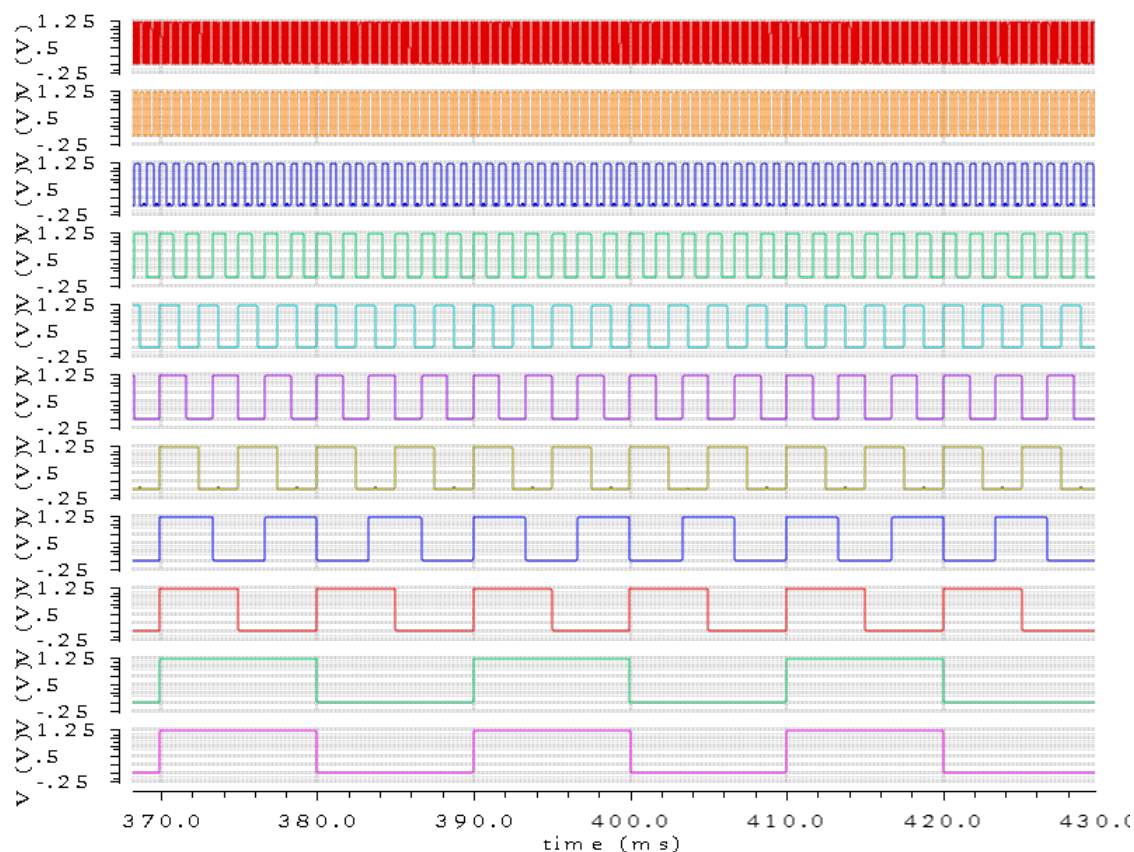


Figura 3.33: Formas de onda nas diversas saídas do divisor N.

A seleção destas frequências é feita através de um multiplexador de 10:1, podendo assim ser usados 4 *bits* em  $f_{sel}$ , sendo atribuídos os valores deste sinal respetivamente com a frequência, por exemplo, para frequência  $f_{Nout} = f_3$  o valor é  $f_{sel} = 3$ .

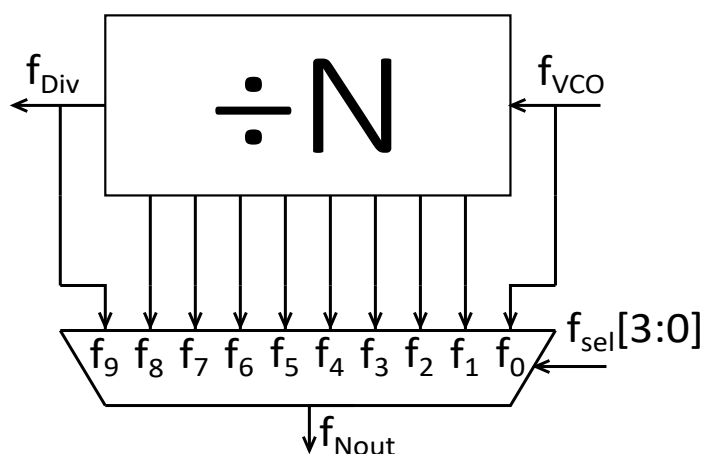


Figura 3.34: Circuito da seleção das frequências à saída.

A não reprogramação do bloco N permite obter as frequências em 3.9 sem ser necessário esperar que a PLL atinja regime estacionário para um novo valor de N, melhorando-se assim o



### 3.2.6 Resultados da implementação final

Por forma a ilustrar e caracterizar o funcionamento da PLL, foram feitas várias simulações usando o simulador Cadence Spectre®. Os casos apresentados correspondem aos da simulação para o caso TT (*typical*), à temperatura de 25°C e  $V_{dd} = 1,2\text{ V}$ , isto é, um caso nominal sem variações PVT.

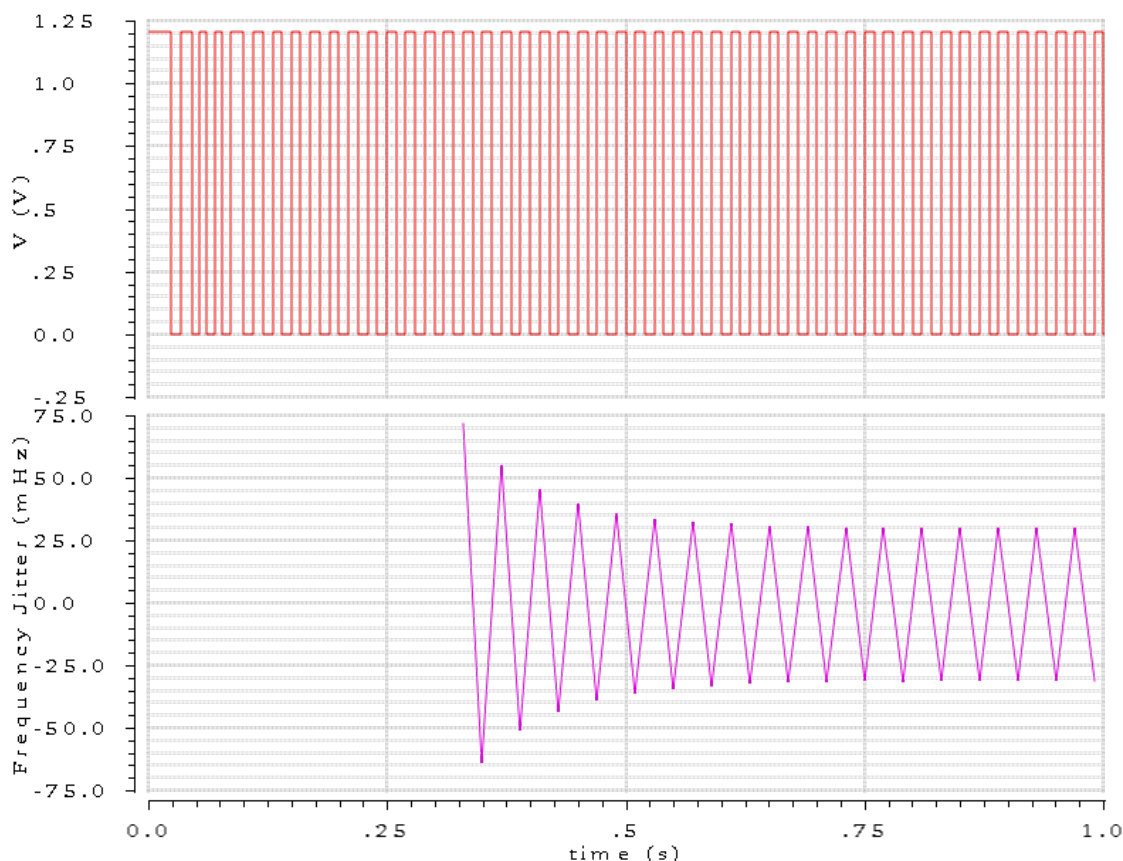


Figura 3.37: *Jitter* de frequência do sinal ( $f_{div}$ ) de 50 mHz à entrada do detetor de fase.

O requisito para que o erro seja menor que 1% em regime estacionário para as frequências entre 600 Hz e 50 mHz pode ser verificado usando a ferramenta do diagrama de olho.

As figuras 3.37 e 3.38 mostram que, à frequência de 50 Hz, a oscilação de frequência e a incerteza da ocorrência do instante de transição satisfazem o requisito de se obter um erro no valor da frequência inferior a 1%, em regime estacionário. É possível observar que a esta frequência o erro máximo é de  $\approx 50\text{ mHz}$ . A satisfação deste requisito é também verificada na figura 3.39 no que diz respeito à incerteza na ocorrência do instante de transição à frequência de 600 Hz.

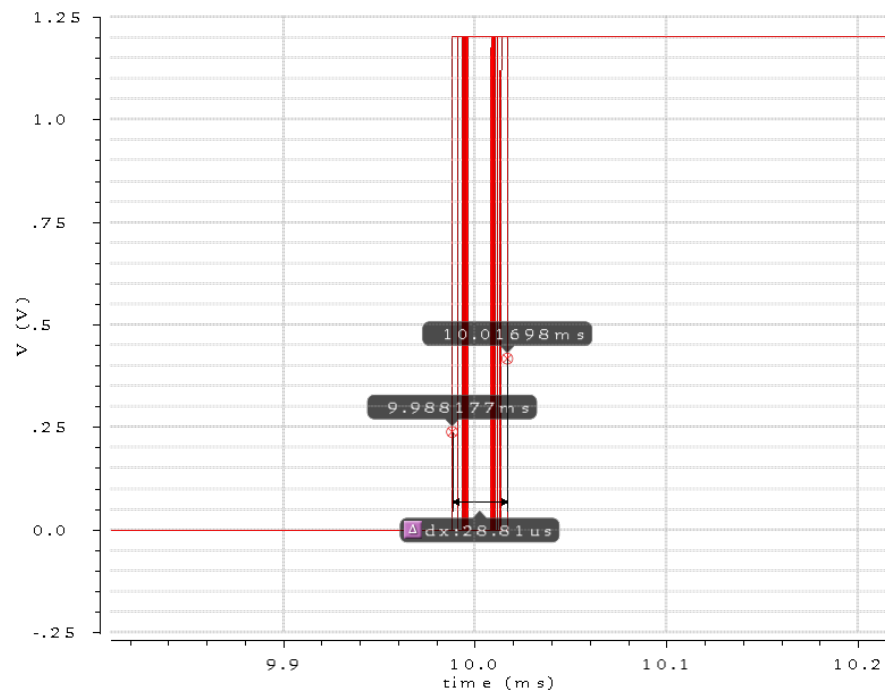


Figura 3.38: Incerteza do instante de ocorrência da transição à frequência de 50 Hz.

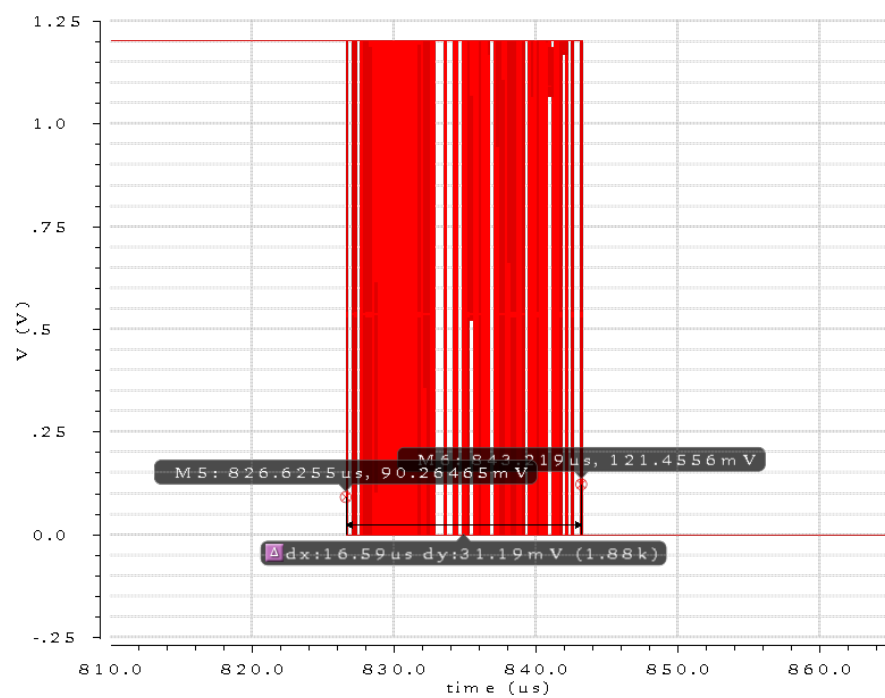


Figura 3.39: Incerteza do instante de ocorrência da transição à frequência de 600 Hz.

O funcionamento do oscilador na resposta ao degrau é ilustrado na figura 3.41.

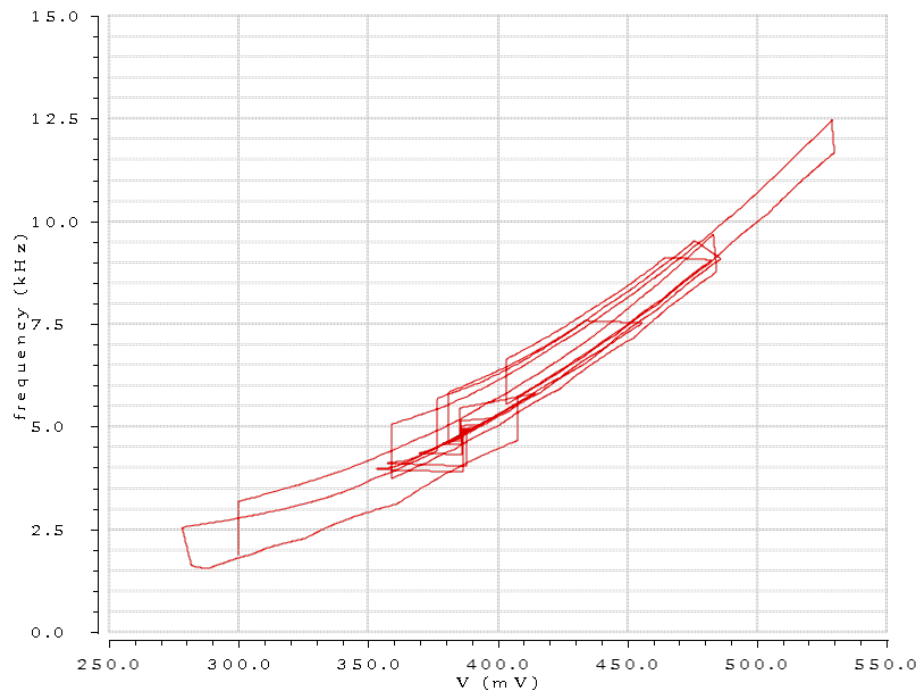


Figura 3.40: Resposta transitória do oscilador ao degrau.

Nesta simulação é definida como condição inicial uma tensão de 300 mV para o valor à saída do filtro, por forma a que o VCO consiga oscilar, sendo que para valores inferiores ( $\approx 200$  mV) o transistor  $M_1$  da figura 3.28, que funciona como fonte de corrente, entra em corte. É também definida a condição inicial de 0 V à saída do VCO por forma a reproduzir o efeito de ligar o circuito, criando a excitação inicial para oscilação.

É possível verificar na figura 3.41 a observação feita em 3.1.5 acerca das limitações do simulador CppSim perante a resposta ao degrau para limites mais estritos das estabilidade. A implementação que resulta do dimensionamento do  $K_{vco} = 65$  kHz/V, apresenta uma resposta instável na figura 3.19 para uma variação de frequência do sinal de entrada, a qual, e para uma variação semelhante, apresenta nas figuras 3.41 e 3.42 uma resposta estável. Esta último comportamento é mais correta dado corresponder a uma simulação mais rigorosa e próxima daquela que resulta do circuito real.

Na figura 3.40 é mostrado o percurso realizado pela saída do oscilador para a resposta transitória apresentada na figura 3.41. É possível observar que o oscilador não segue uma curva linear para as mesmas variações de frequência. Tal comportamento deve-se aos efeitos de carga e descarga da capacidade  $C_L$  — presente na célula de atraso apresentada na figura 3.28 — assim como a outras não linearidades [51].



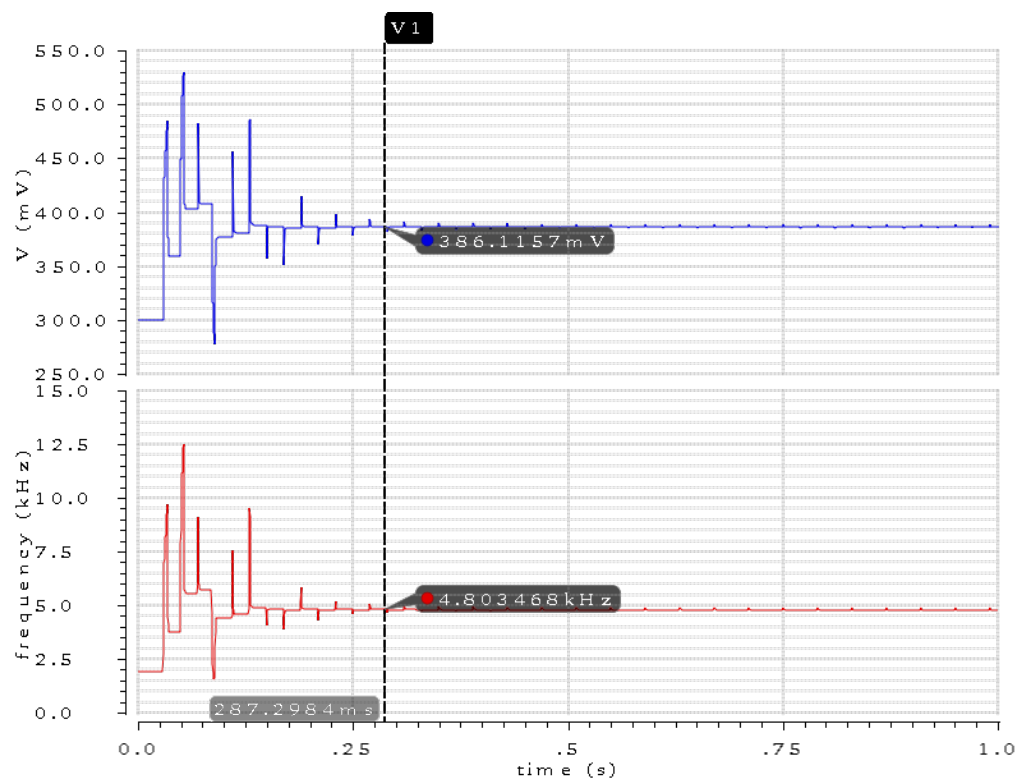


Figura 3.41: Resposta ao degrau do sistema implementado 3.32 à saída do VCO.

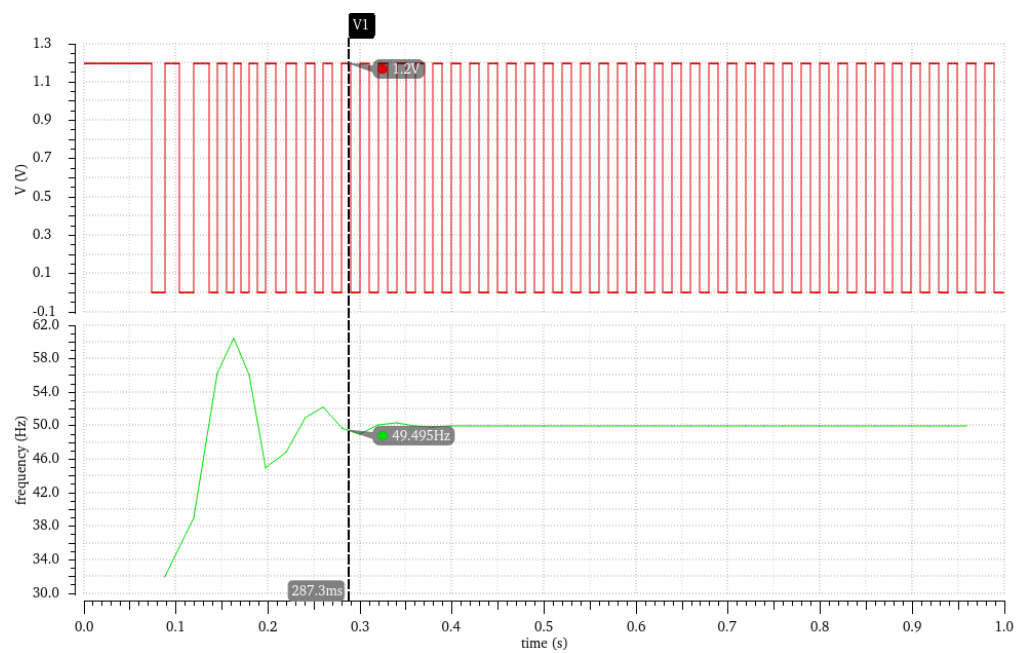


Figura 3.42: Resposta ao degrau do sistema implementado 3.32 visto na frequência de 50 Hz.

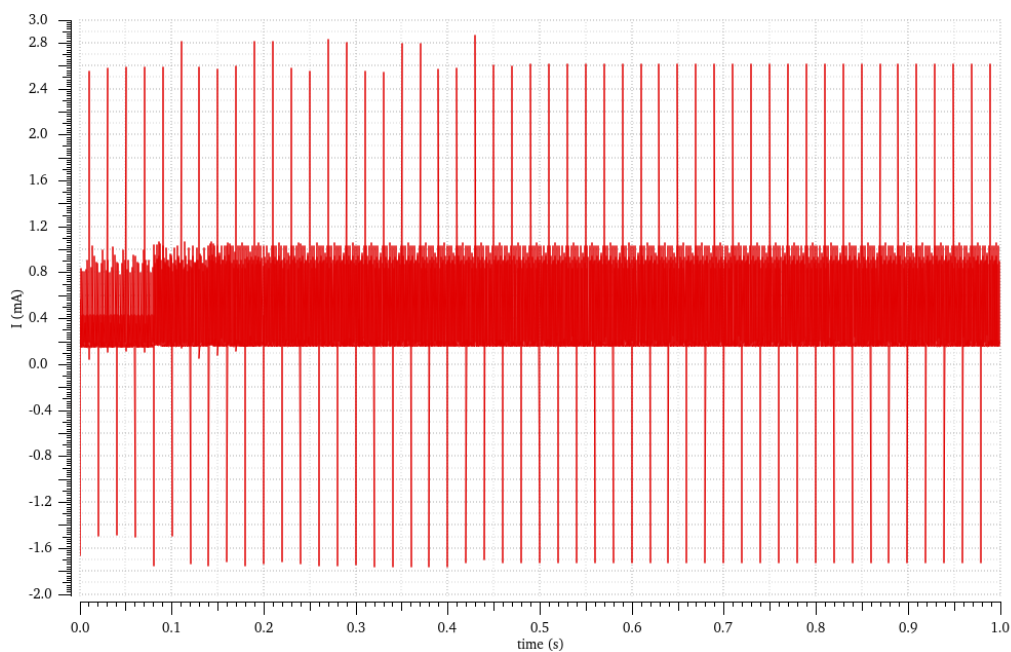


Figura 3.43: Consumo corrente visto da fonte tensão  $V_{dd}$ .

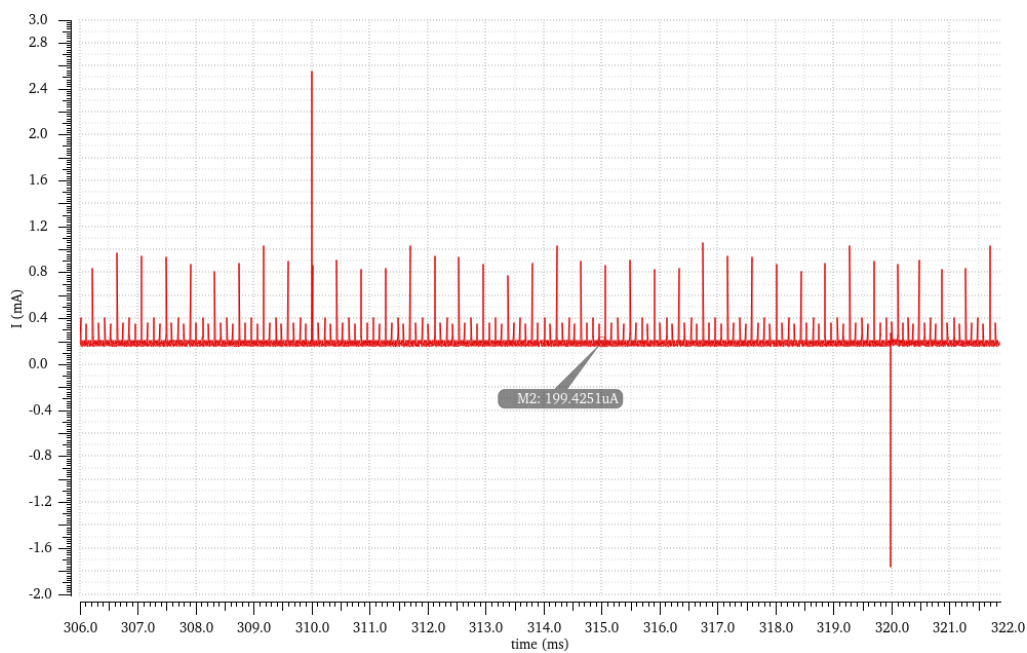


Figura 3.44: Consumo corrente visto da fonte tensão  $V_{dd}$  (ampliada).

Através das figuras 3.43 e 3.44 é possível obter a potência média do sistema da PLL sendo dada para  $T=1$  s e  $V_{dd}=1,2$  V:

$$P_{avg} = \frac{V_{dd}}{T} \int_0^T I_{avg}(t) dt \approx 223,1 \mu W \quad (3.43)$$

## Capítulo 4

# Calibração e Testabilidade

No presente capítulo é proposto um algoritmo de calibração baseado nas metodologias usadas atualmente em aprendizagem automática e reconhecimento de padrões 2.3. Para calibrar uma PLL é necessário avaliar quais os fatores que afetam o seu comportamento, por meio de análise das possíveis variabilidades, isto, é uma análise de sensibilidade do circuito a fatores externos, e selecionar onde atuar para se obter a correção necessária. Estes tipos de desvios são conhecidos por PVT e podem ser categorizados de forma distinta, não só pelo seu efeito mas também pela maneira como influenciam o correto funcionamento do sistema. Podem ser classificados em três grupos distintos, no que diz respeito ao período de tempo envolvido:

- Efeitos de variações de processo, envelhecimento e outros tipos de variações lentas podem ser consideradas variações do tipo estáticas, e podem ser corrigidas tomando constantes de tempo longas.
- Efeitos da temperatura são geralmente dinâmicos e lentos, dado a curva do aumento/ decréscimo do fator da temperatura.
- Efeitos de variações de tensão e outras interferências transitórias são consideradas variações dinâmicas e rápidas, afetando o sistema uma única vez ou durante um curto espaço de tempo do funcionamento da PLL.

De forma a saber como estes efeitos afetam o circuito é necessário fazer um levantamento de fatores de sensibilidade e identificar quais os casos mais extremos. Para isso:

- é feita uma análise *a priori* envolvendo estudos estatísticos (Monte Carlo);
- ajustar estas variações segundo uma curva padronizada (*GPR fit*);
- identificar quais os piores casos.

Este procedimento envolve uma análise de variações de fatores externos e exige a recolha de uma grande coleção de dados para se conseguir fazer uma previsão precisa de todos os efeitos, mesmo os mais extremos.

## 4.1 Metodologia

Tomando como exemplo o caso de estudo [14] apresentado na secção 2.3.2, é possível seleccionar as melhores *features* usando o algoritmo de *feature selection* descrito em 2.3.1. No entanto, dada a complexidade do sistema da PLL e o longo tempo associado à sua simulação (considerando variações PVT), e por forma a obter um conjunto de dados suficientemente grande, a obtenção de um subconjunto de *features* ótimo é uma tarefa computacionalmente exigente, de certo modo incompatível com o tempo de realização disponível para este trabalho.

Deste modo foi feito uma pre-selecção das *features* tomando uma análise dedutiva das equações do funcionamento de cada bloco da PLL por forma a ser possível encontrar o ponto de atuação para calibração.

Sendo assim, recorrendo à análise da função em malha aberta do sistema, é possível decidir quais os parâmetros mais sensíveis a variações PVT, assim como também a facilidade e a complexidade deste ajuste 2.3.1, baseando-se assim em parte com a literatura revista no capítulo 2. Considerando os três blocos constituintes da PLL, pode escrever-se,

$$G(s, P, V, T) = \frac{K_{pd_{alt}}(P, V, T)K_{vco}(P, V, T)F(s, P, T)}{s} \quad (4.1)$$

onde se põe em evidência que o CP e o VCO são os blocos mais suscetíveis de sofrerem efeitos de todas as possíveis causas consideradas. Assumindo que as variações PVT têm maior impacto no ganho é possível então decidir que os fatores mais sensíveis são  $K_{pd_{alt}}$  — devido a  $I_{cp}$  — e  $K_{vco}$ . Uma vez que é a tensão de controlo do VCO, a tensão à saída do filtro, dos parâmetros mais críticos para a estabilidade do circuito. Expressando-a em função do erro de fase, vem

$$\Delta V_{ctrl} = K_{filter}(\pm(I_{cp} \pm I_{cp}(P, V, T)) \cdot \Delta T_{up/down}) \quad (4.2)$$

Como esta tensão é amplificada por  $K_{vco}$ , vê-se que VCO tem um peso significativo no processo de manutenção da estabilidade da PLL, pelo que manter o oscilador calibrado dentro de limites de sintonia bem estabelecidos, requer um controlo direto dos desvios do  $K_{vco}$ .

Por forma a calibrar a PLL é então feita uma calibração por forma a normalizar os parâmetros de estabilidade dado a dependência de  $\zeta(K_{vco})$  e de  $\omega_n(K_{vco})$ .

O diagrama de blocos apresentado na figura 4.1 apresenta a proposta da implementação de autocalibração do sistema da PLL.

O bloco da memória juntamente com o bloco comparador e decisor são os bloco centrais do estudo apresentado na secção 4.1.1, é neste que são guardados os resultados da análise do sistema obtidos.

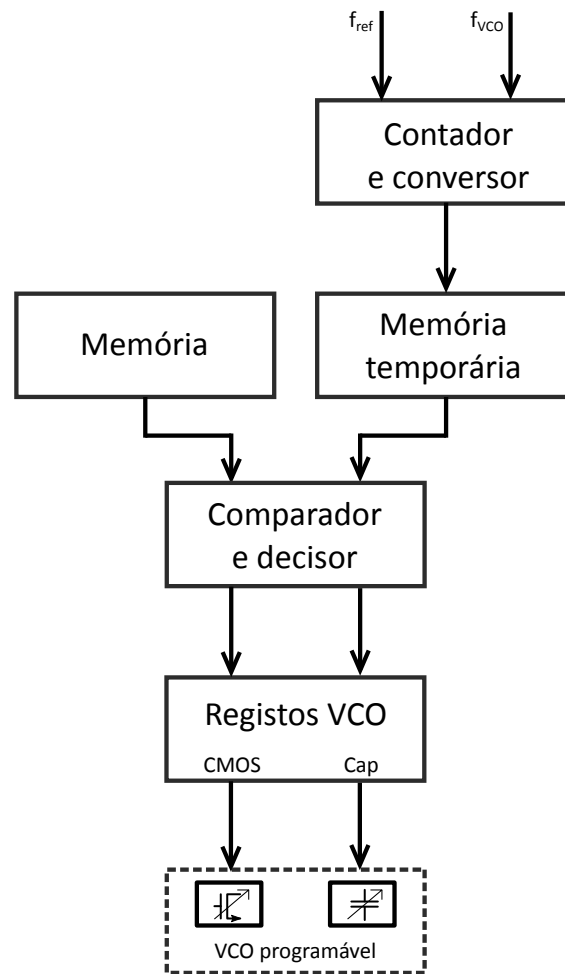


Figura 4.1: Diagrama de blocos generalizado da implementação proposta.

A restante constituição do diagrama de blocos segue a metodologia apresentada em 4.1.2, utilizando algumas técnicas descritas no capítulo 2 e propondo uma metodologia de calibração para implementação na PLL desenhada no capítulo 3.

#### 4.1.1 Análise *a priori*

Considerando então como objeto de estudo apenas os efeitos PVT no  $K_{vco}$  procedeu-se a uma análise *a priori* do VCO. Para tal foi isolado o oscilador do circuito e feita uma grande quantidade de simulações de funcionamento, de modo a obter um conjunto de dados com variações de processo e de seguida variações com todos os fatores PVT.

A simulação consiste num varrimento dos valores possíveis de  $V_{ctrl}$ , obtendo para cada  $V_{ctrl}$  um valor de frequência correspondente à saída do oscilador. Esta simulação compreendeu uma simulação Monte Carlo e simulações de casos extremos (*corners*) [52], ambas realizadas no ambiente Spectre® da Cadence. Nas simulações para análise de Monte carlo foi tomado o valor de  $\sigma = 3$ , de modo a cobrir um valor aceitável do rendimento do fabrico, isto é 99,7% dos casos de variação de processo.

Com o conjunto de pontos obtido, denominado por dados de treino (*train data*) [53] por ser usado para definir um modelo de predição, obtém-se uma curva do comportamento mais provável, permitindo assim definir o  $K_{vco}$  mais provável assim como delimitar os respectivos limites de funcionamento aceitáveis.

Este modelo de treino é baseado num processo de regressão gaussiana (GPR)[31]. Este consiste numa abordagem probabilística baseada em núcleos não-paramétricos, que retorna o resultado dos valores em  $V_{ctrl_i}$  para predições em  $f_{vco_i}$  com erro  $\varepsilon_i$ .

Apesar de os valores estarem estocasticamente distribuídos entre si, devido às variações impostas pela simulação de Monte Carlo, os pontos sequenciais — seguindo o gradiente de aumento do ganho do  $K_{vco}$  com  $V_{ctrl}$  — seguem uma distribuição gaussiana sendo portanto a adaptação deste modelo uma ferramenta muito precisa, em relação ao erro perante o valor nominal de  $K_{vco}$  [14].

$$f_{vco_i} = f(V_{ctrl_i}) + \varepsilon_i \quad (4.3)$$

Sendo os dados de treinos compostos por pares  $(y_i, x_i) \rightarrow (f_{vco_i}, V_{ctrl_i})$ .

$$f \sim \text{GP}(m(x_i), K(x_i, x_j)) \quad (4.4)$$

$$\begin{pmatrix} y_i \\ y_{i*} \end{pmatrix} \sim \mathcal{N}\left(0, \begin{pmatrix} K & K_*' \\ K_* & K_{**} \end{pmatrix}\right) \quad (4.5)$$

onde  $K$  é a matriz de covariância de  $f_{vco}$ ,  $K_*$  é o vetor de covariância entre um novo ponto  $f_{vco_*}$  e os pontos de  $f_{vco}$ , e  $K_{**}$  é o ruído inerente à medição assumindo uma média normalizada de 0 [54].

$$p(y_{i*} | y_i) \sim \mathcal{N}(K_* K^{-1} y_i, K_{**} - K_* K^{-1} K_*') \quad (4.6)$$

Com o propósito de selecionar o núcleo mais apropriado para a distribuição em causa, decidiu-se aplicar um núcleo não-paramétrico em exponencial quadrático (SE).

$$K_{SE}(x, x') = \sigma^2 \exp\left(-\frac{(x - x')^2}{2\ell^2}\right) \quad (4.7)$$

em que  $\sigma^2$  corresponde à variância e  $\ell$  à escala de comprimento (*Lengthscale*) [31].

Também foi estudada a opção de usar um núcleo Matérn definido por

$$C_v(d) = \sigma^2 \frac{2^{1-\nu}}{\Gamma(\nu)} \left(\sqrt{2\nu} \frac{d}{\rho}\right)^\nu K_\nu\left(\sqrt{2\nu} \frac{d}{\rho}\right) \quad (4.8)$$

onde  $d$  é a função euclidiana da distância,  $\Gamma$  a função Gamma,  $K_\nu$  a função modificada de Bessel de segunda geração,  $\rho$  e  $\nu$  parâmetros positivos, sendo usado neste caso os valores mais populares de  $\nu = \frac{3}{2}$  e  $\nu = \frac{5}{2}$ .

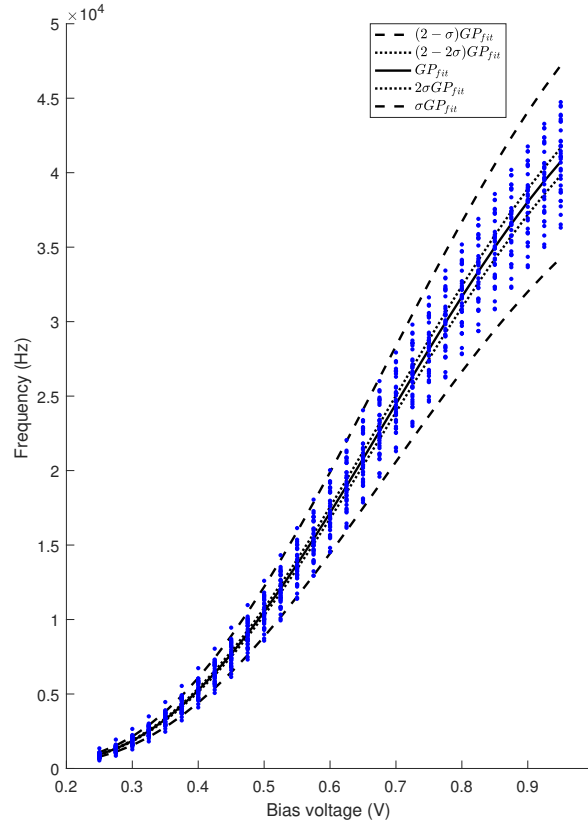


Figura 4.2: Variações de processo para 1015 pontos de dados.

No entanto os resultados foram muito semelhantes aos obtidos com o núcleo 4.7, tendo sido portanto usado o núcleo mais simples baseado na exponencial quadrática.

As curvas na figura 4.2 a traço interrompido, são definidas pelo resultado do ajuste feito pela curva da predição do GPR ( $GP_{fit}$ ), multiplicando-as por  $[2-\sigma \ 2-2\sigma \ 2\sigma \ \sigma]$ , para  $\sigma=84.13\%$  e  $2\sigma=97.73\%$ . Estas correspondem a limites pré-estabelecidos aos quais — em valores nominais — a PLL apresenta a melhor resposta após a resposta transitória, melhorando o *jitter* em frequência e relaxando os valores para a frequência natural  $\omega_n$  do sistema e tempo de estabelecimento  $t_s$ .

Quanto a variações de temperatura, foi feito isoladamente um varrimento de valores de temperatura de modo a identificar qual o efeito provocado na frequência do oscilador. Estes são ilustrados na figura 4.3 tal como é proposto em [14].

De seguida foi feita a simulação considerando todos os efeitos PVT (figura 4.4) tomando para os valores de tensão da alimentação  $V_{DD_{dc}} = V_{DD_{nom}} + V_{var_i}$ , em que  $V_{DD_{nom}}$  é a tensão nominal de 1,2 V e  $V_{var_i}$  o conjunto  $[-100 \ 0 \ 100]$  mV.

Os valores das curvas que apresentam os extremos das variações representam os máximos e mínimos  $K_{vco}$  cobertos nesta calibração, sendo os restantes dados valores de cantos com baixa probabilidade de ocorrência.

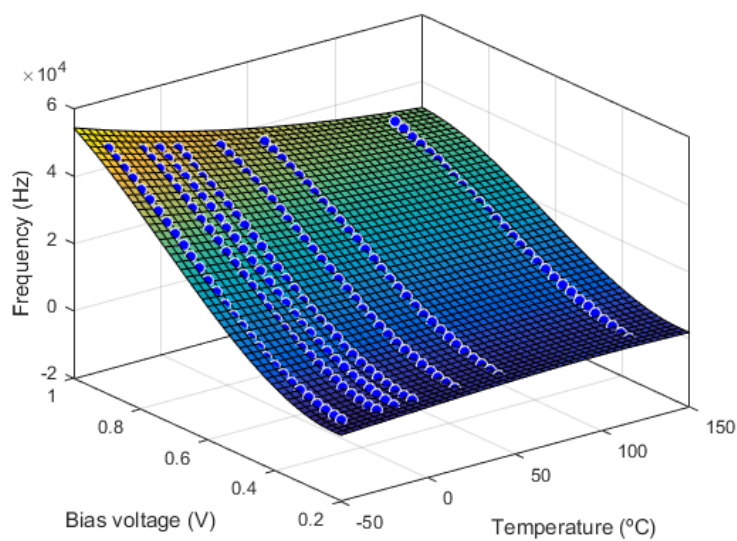


Figura 4.3: Impacto das variações de temperatura para o intervalo entre  $-40^{\circ}\text{C}$  e  $125^{\circ}\text{C}$ .

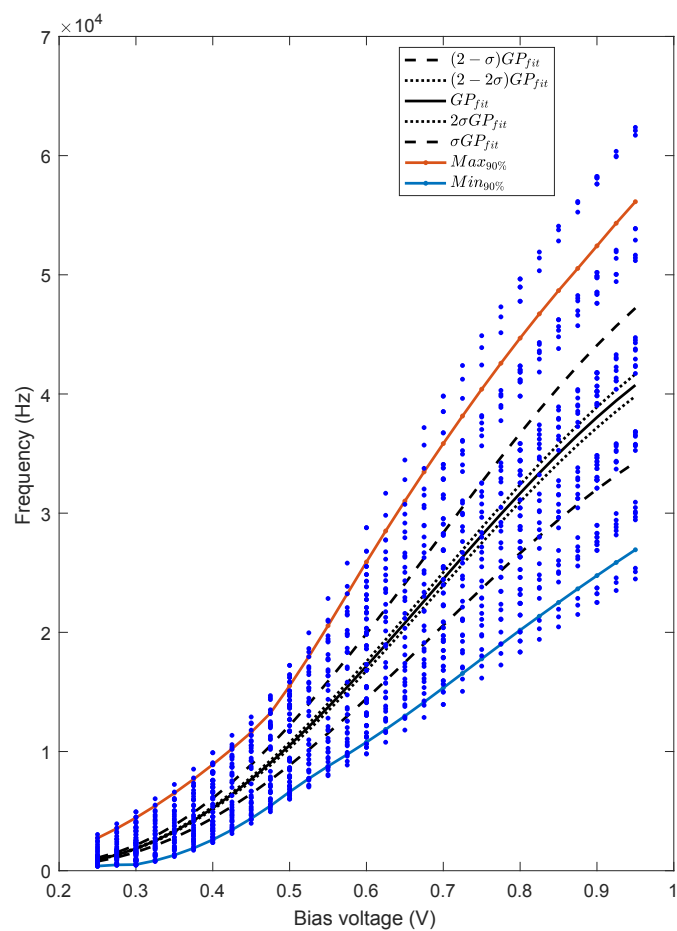


Figura 4.4: Variações de PVT para 1015 pontos de dados.



### 4.1.2 Análise *a posteriori*

Com dos valores obtidos em 4.1.1 é definido um conjunto de vetores para cada curva dos intervalos em 4.4. De modo a minimizar o conjunto de dados a serem usados, foram definidos os pontos a partir de  $V_{ctrl} = 0,4 \text{ V}$  até  $V_{ctrl} = 0,8 \text{ V}$  com passo de  $0,1 \text{ V}$ , obtendo-se o seguinte vetor simplificado:

$$f_i(x_j) = y_i \rightarrow f_{vco_i} \begin{bmatrix} V_{ctrl_1} \\ V_{ctrl_2} \\ V_{ctrl_3} \\ V_{ctrl_4} \\ V_{ctrl_5} \end{bmatrix} = \begin{bmatrix} y_{i1} \\ y_{i2} \\ y_{i3} \\ y_{i4} \\ y_{i5} \end{bmatrix} \quad (4.9)$$

com  $i = 1, 2 \dots 6$  correspondente a cada curva e  $j = 1, 2 \dots 5$  a cada valor de  $V_{ctrl}$ , obtendo-se assim o valor  $y_{ij}$  de frequência de oscilação. Atendendo esta atribuição é possível obter um conjunto de pontos que definem as curvas do gráfico 4.4.

$$\begin{aligned} f_{Max} &= \begin{bmatrix} f_{Max90\%}(V_{ctrl_1}) \\ f_{Max90\%}(V_{ctrl_2}) \\ f_{Max90\%}(V_{ctrl_3}) \\ f_{Max90\%}(V_{ctrl_4}) \\ f_{Max90\%}(V_{ctrl_5}) \end{bmatrix} & f_{H\sigma} &= \begin{bmatrix} f_{(2-\sigma)}(V_{ctrl_1}) \\ f_{(2-\sigma)}(V_{ctrl_2}) \\ f_{(2-\sigma)}(V_{ctrl_3}) \\ f_{(2-\sigma)}(V_{ctrl_4}) \\ f_{(2-\sigma)}(V_{ctrl_5}) \end{bmatrix} & f_{H2\sigma} &= \begin{bmatrix} f_{(2-2\sigma)}(V_{ctrl_1}) \\ f_{(2-2\sigma)}(V_{ctrl_2}) \\ f_{(2-2\sigma)}(V_{ctrl_3}) \\ f_{(2-2\sigma)}(V_{ctrl_4}) \\ f_{(2-2\sigma)}(V_{ctrl_5}) \end{bmatrix} \\ f_{L2\sigma} &= \begin{bmatrix} f_{2\sigma}(V_{ctrl_1}) \\ f_{2\sigma}(V_{ctrl_2}) \\ f_{2\sigma}(V_{ctrl_3}) \\ f_{2\sigma}(V_{ctrl_4}) \\ f_{2\sigma}(V_{ctrl_5}) \end{bmatrix} & f_{L\sigma} &= \begin{bmatrix} f_{(\sigma)}(V_{ctrl_1}) \\ f_{(\sigma)}(V_{ctrl_2}) \\ f_{(\sigma)}(V_{ctrl_3}) \\ f_{(\sigma)}(V_{ctrl_4}) \\ f_{(\sigma)}(V_{ctrl_5}) \end{bmatrix} & f_{Min} &= \begin{bmatrix} f_{Min90\%}(V_{ctrl_1}) \\ f_{Min90\%}(V_{ctrl_2}) \\ f_{Min90\%}(V_{ctrl_3}) \\ f_{Min90\%}(V_{ctrl_4}) \\ f_{Min90\%}(V_{ctrl_5}) \end{bmatrix} \end{aligned} \quad (4.10)$$

Seguindo a metodologia descrita em 2.23b, é decidido o tipo de calibração a efetuar aplicando uma seleção de quais os fatores que influenciam o funcionamento do oscilador, assim como sabendo de que modo é que a alteração destes provoca um aumento ou um decréscimo do valor do  $K_{vco}$ .

Atendendo à equação de funcionamento do oscilador 3.41, podemos observar que o aumento da corrente de cada célula do inversor *current starved* provoca um aumento diretamente proporcional na frequência. De igual modo vê-se que o aumento da capacidade em cada célula de atraso provoca um decréscimo na frequência, sendo portanto, inversamente proporcional.

Tomando os limites definidos em 4.1.1 para os valores máximos e mínimos, é calculado qual o valor a acrescentar em cada célula de atraso para que quando conectados ao circuito, o valor do  $K_{vco}$  possa ser "transportado" para o valor mais provável da simulação de Monte Carlo, isto é, que o vetor de frequências  $f_n$  após aplicada a calibração, contenha os pontos dentro da área definida

pelas frequências:

$$f_{H2\sigma} \geq f_n \geq f_{L2\sigma} \quad (4.11)$$

Podem-se assim implementar dois processos de calibração:

- Calibração para aumento do  $K_{vco}$  : De modo a aumentar a corrente em cada andar do inversor *current starved*, é colocado em paralelo com cada transístor do espelho de corrente um banco de 3 transístores com as proporções de  $\frac{W}{L} \cdot [\frac{1}{8} \frac{2}{8} \frac{5}{8}]$  em relação ao  $\frac{W}{L}$  previamente desenhado. Caso os valores de frequência se encontrem no intervalo:

$$f_{L2\sigma} > f_n \geq f_{Min} \quad (4.12)$$

Neste intervalo a PLL tem tendência a aumentar o tempo de estabelecimento como já visto em na figura 3.12.

- Calibração para diminuição do  $K_{vco}$  : Para aumentar a capacidade é feito o mesmo processo, colocando em paralelo com cada capacidade  $C_L$  um banco de 3 capacidades com as proporções de  $C_L \cdot [\frac{1}{8} \frac{2}{8} \frac{5}{8}]$ . Caso os valores de frequência se encontrem no intervalo:

$$f_{Max} \geq f_n > f_{H2\sigma} \quad (4.13)$$

O aumento do  $K_{vco}$  provoca um aumento do *jitter* em frequência na PLL, sendo que para o mesmo  $V_{ctrl}$  o salto em frequência é maior, provocando uma instabilidade no comportamento [9, 55].

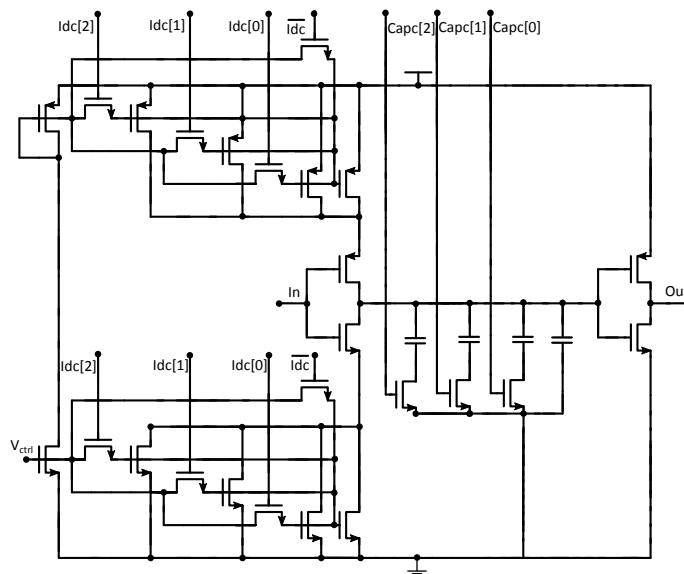


Figura 4.5: Célula de atraso programável proposta para calibração do VCO.

A cada um destes elementos é conectado um transístor que funciona como interruptor, podendo ser ligado ou desligado consoante o resultado do algoritmo. A configuração apresentada é portanto

uma combinação de 6 bits de controlo permitindo assim 64 combinações diferentes para o ajuste pretendido, sendo 32 destas para um  $K_{vco}$  alto e as restantes 32 para um  $K_{vco}$  baixo.

$$controller(Idc, Capc) = \begin{bmatrix} x_{00} & x_{01} & \dots & x_{07} \\ x_{10} & x_{11} & \dots & x_{17} \\ \vdots & \vdots & \ddots & \vdots \\ x_{70} & x_{71} & \dots & x_{77} \end{bmatrix} \quad (4.14)$$

Atendendo a esta análise é obtido o seguinte algoritmo iterativo apresentado na figura 4.6, o qual define um conjunto de zonas de operação por forma a criar um conjunto de padrões para calibrar consoante a metodologia apresentada. As curvas da figura 4.4 e os vetores 4.10 definem zonas de ganho ao quais o oscilador se pode encontrar devido aos efeitos PVT, sendo estas dadas por  $Z_i$ , com  $i=1, \dots, 5$ :

- $Z_1 : f_n > f_{Max} \vee f_n < f_{Min}$
- $Z_2 : f_n \geq f_{L2\sigma} \wedge f_n \leq f_{H2\sigma}$
- $Z_3 : f_n > f_{H2\sigma} \wedge f_n \leq f_{Max}$
- $Z_4 : f_n \geq f_{Min} \wedge f_n < f_{L2\sigma}$
- $Z_5 : f_n \geq f_{L\sigma} \wedge f_n \leq f_{H\sigma}$

O fluxograma da figura 4.6 ilustra como é feita a implementação do algoritmo de calibração, onde:

- Setup  $f_n$ : Conta a frequência para cada  $V_{ctrl_i}$ , inicializando o vetor de dados de teste  $f_n$ ;
- $K_{vco}$  zona: Compara cada ponto do vetor  $f_n$  em relação às zonas definidas em  $Z_i$ ;
- Caso A: Caso  $Idc$  não possa ser incrementado, é combinado o ajuste usando  $Idc$  e  $Capc$  em simultâneo;
- Caso B: Caso  $Capc$  não possa ser incrementado, é combinado o ajuste usando  $Idc$  e  $Capc$  em simultâneo;

Para efetuar a contagem de frequência é utilizada a entrada  $F_{ref}$  como referência para o cálculo, tomando assim a resolução como

$$R_s = \frac{F_{vco}}{F_{ref}} \quad (4.15)$$

$$\lim_{F_{ref} \rightarrow 0} R_s = \lim_{F_{vco} \rightarrow \infty} R_s \Leftrightarrow R_s = \infty$$

onde o erro de 1 *tick* na contagem corresponde a 1 ciclo do relógio da referência, daí que para que exista uma maior precisão na contagem é necessário garantir que a frequência de oscilação do oscilador seja muito superior à da referência [9]. Este princípio de medição baseia-se nos apresentados na secção 2.2, sendo conhecida por conversão frequência-digital (FDC).

A cada iteração do algoritmo correspondem 6 ciclos do relógio de referência, sendo 5 para leitura do estado atual do  $K_{vco}$  e inicialização do vetor  $f_n$ , e 1 ciclo para estabelecer as relações de comparação das zonas assim como a ativação dos valores correspondentes de  $I_{dc}$  e  $Capc$ .

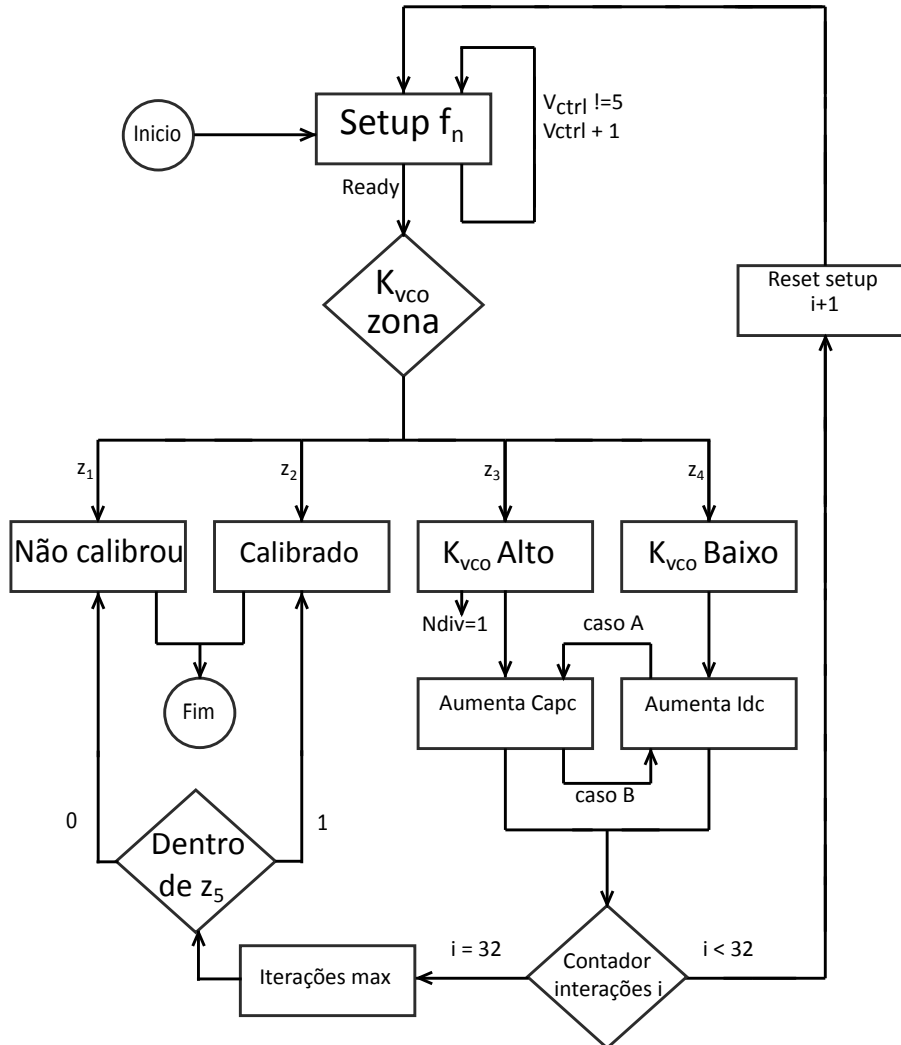


Figura 4.6: Fluxograma do algoritmo de calibração.

A decisão feita em "Dentro de  $Z_5$ " ocorre quando o algoritmo atinge um máximo de iterações, caso o algoritmo não consiga convergir mas garante que os pontos de  $f_n$  se encontrem dentro da zona  $Z_5$ , é assumida uma calibração aceitável do  $K_{vco}$ . O resultado  $Ndiv=1$  serve para compensação de um  $K_{vco}$  alto, pelo que eleva a zona de funcionamento do VCO para um valor mais alto, logo neste caso, é forçada a saída a 9600 Hz — a custo de inclusão de mais um divisor — permitindo assim que não se aproxime de um  $V_{ctrl}$  muito baixo, a ponto de forçar o corte do transistor que opera como fonte de corrente. O limite de 32 iterações vem de acordo com as combinações máximas que determinam uma convergência do algoritmo, admitindo que sem esta condição se o algoritmo não convergisse entraria num ciclo infinito.

A maior vantagem da análise *a priori* decorre de se limitar e definir em que zonas a calibração

é feita, sem considerar valores que inicialmente não tinham uma limitação, como o caso do valor associado às capacidades  $Capc$  e aos valores de  $Idc$ . Ao criar zonas que estabelecem uma relação de proporcionalidade direta ao  $K_{vco}$  mais provável de um desvio estático (processo) garante uma calibração mais correta do sistema.

### 4.1.3 Módulos lógico Verilog HDL

De modo a verificar o algoritmo foram criados dois módulos em Verilog HDL, um para emular o comportamento de um oscilador controlado por tensão, e outro um módulo RTL do controlador que efetua a calibração.

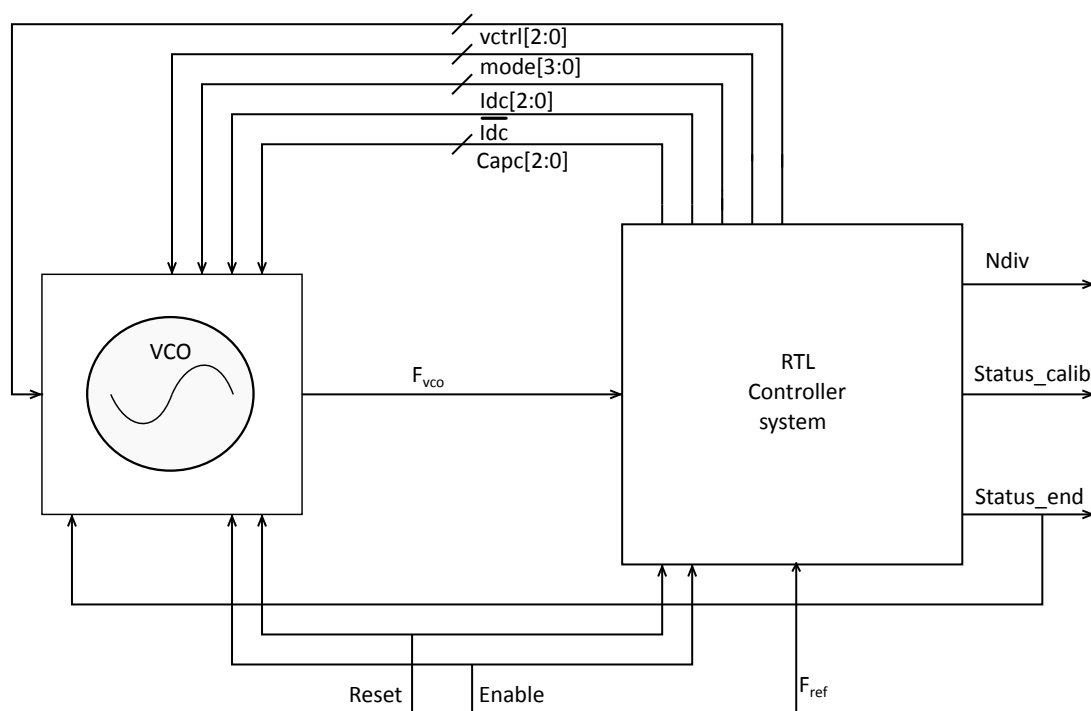


Figura 4.7: Diagrama de blocos Verilog HDL de verificação do processo de calibração.

#### 4.1.3.1 Módulo do oscilador controlado por tensão

O módulo do VCO consiste num bloco RTL não sintetizável que tem as seguintes portas:

- Entradas :

- |          |                |                  |                  |
|----------|----------------|------------------|------------------|
| – Enable | – $Vctrl[2:0]$ | – $Idc[2:0]$     | – $mode[3:0]$    |
| – Reset  | – $Capc[2:0]$  | – $Inverted\_in$ | – $Status_{end}$ |

- Saídas :

- $F_{vco}$

Este módulo funciona como um modelo matemático que gera um valor de saída em "1" durante o tempo correspondente a metade do atraso calculado e a restante metade em "0", gerando assim uma frequência desejada de  $\frac{1}{2 \cdot \text{atraso}}$  Hz. O valor do atraso é calculado e atualizado com a resolução de 1 ns, sendo a expressão da frequência a seguinte:

$$f_{vco}(V_{ctrl}, modo, Idc, Capc) = \frac{I(V_{ctrl}, modo)(1 + I_{alt}(Idc))}{2 \cdot 3C_L(1 + C_{alt}(Capc))} \quad (4.16)$$

em que  $C_{alt}$  é o ajuste feito por aumento da capacidade, e  $I_{alt}$  a alteração por acréscimo de corrente, sendo ambos selecionados pelos valores de controlo  $Idc$  e  $Capc$  respetivamente. Caso os valores de  $C_{alt}$  e  $I_{alt}$  sejam nulos, a frequência da saída  $F_{vco}$  segue os valores pré definidos pelo valor de  $V_{ctrl}$  selecionado.

De modo a linearizar o modelo foi feita uma aproximação do funcionamento em 4.4 para seguir a equação em 4.16.

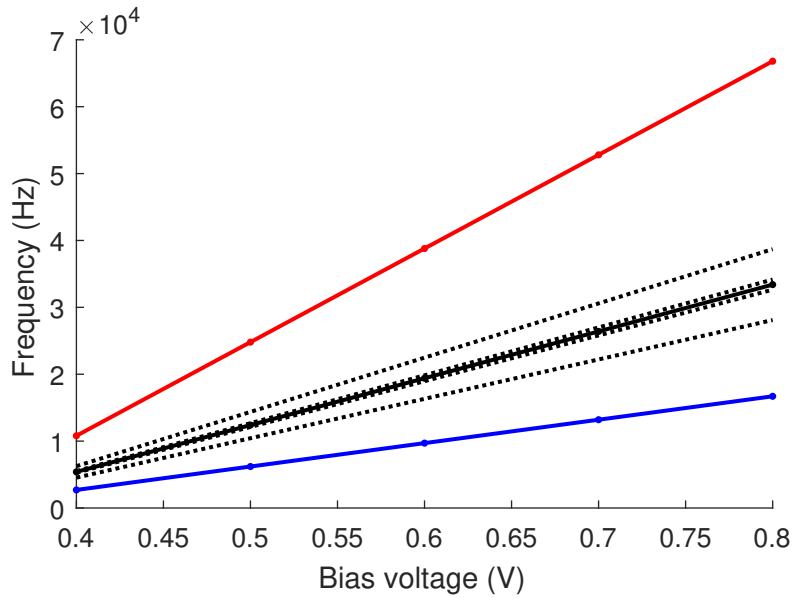


Figura 4.8: Linearização  $K_{vco}$  em relação ao 4.4.

#### 4.1.3.2 Módulo do controlador da calibração

O módulo do controlador da calibração consiste num bloco RTL sintetizável que tem as seguintes portas:

- Entradas :

–  $F_{vco}$

– Reset

– Enable

–  $F_{ref}$

- Saídas :

- Ndiv
- Status\_end
- Capc[2:0]
- Inverted\_in
- Status\_calib
- Vctrl[2:0]
- Idc[2:0]
- mode[3:0]

Este módulo funciona como controlador dos valores a alterar para as entradas do oscilador Idc e Capc. Para tal é feita a comparação dos resultados das equações 4.12 e 4.13 com valores previamente guardados em memória. Como referido em 4.1.2, o uso de 5 pontos de leitura de frequência permite a redução do tamanho da memória para 5 registos para cada curva, sendo esta agrupada em 6 blocos definidos pelos vetores em 4.10.

Seguindo os passos na figura 4.6, é inicialmente feita uma leitura dos valores de  $f_{vco}$ , para os  $V_{ctrl}$  respetivos, isto é forçando a entrada com o vetor definido na equação 4.9.

Os valores guardados na memória representativos das zonas  $Z_i$  estão definidos em função de  $F_{ref}$ , isto é para efetuar a conversão da frequência efetivamente lida é necessário multiplicar o número de *ticks* por  $F_{ref}$ .

#### 4.1.3.3 Síntese módulo controlador

Como referido, o bloco do controlador ilustrado na figura 4.7 é um bloco sintetizável que, usando o ambiente Xilinx ISE®, permite obter uma solução que apresenta as condições necessárias para ser implementado em FPGA/ASIC.

Durante o processo de síntese é possível decidir qual é o objetivo da otimização da implementação, isto é, se o objetivo é otimizar o circuito por forma a diminuir a área total do circuito, ou se o objetivo é que o circuito otimizado seja mais rápido, atendendo que cada uma das implementações se apresenta a contrapartida da outra [56]. Como é enunciado nos requisitos do sistema em 3.1.2, a minimização da área é prioritária.

Tabela 4.1: Resultado sumário da síntese em valores estimados.

Utilização lógica	Quantidade
Número de Registos	130
Número de LUTs	198

É possível ver na tabela 4.2 que o multiplexador de 11 bits 5:1 corresponde ao que efetivamente escolhe as zonas de operação. A quantidade detalhada, correspondente ao número de comparadores resulta da necessidade de comparar o vetor  $f_n$  — no seguimento do diagrama — do passo  $K_{vco}$  zona, como é visto na figura 4.6. O valor correspondente ao número de registos surge de que o mapeamento da memória que contém os vetores 4.10 assim como alguns registos de controlo e de uso como *buffers*/temporários.

Dado que a análise da velocidade máxima do sistema depende do modelo da FPGA e/ou ASIC considerado para a implementação, é feito, a título de exemplo, uma breve descrição do relatório da análise temporal.

O *chipset* usado nesta análise corresponde ao XA7A100T da família Automotive Artix7.

Tabela 4.2: Resultado detalhado da síntese.

Bloco lógico		Quantidade
Somadores		3
	12-bit	1
	3-bit	2
Contadores		2
	3-bit	1
	5-bit	1
Registos		122
	Flip-Flops	122
Comparadores		30
	11-bit >	25
	11-bit $\leq$	5
Multiplexadores		40
	1-bit 2:1	20
	11-bit 5:1	1
	12-bit 2:1	5
	3-bit 2:1	14

- Período mínimo / Frequência máxima : 2,961 ns / 337,735 MHz.
- Tempo mínimo de chegada da entrada antes do relógio de maior frequência (VCO) : 1,778 ns.
- Tempo máximo de espera pela saída em relação ao relógio de maior frequência (VCO) : 1,418 ns.
- Caminho crítico corresponde ao último bit do contador da frequência do VCO.

#### 4.1.3.4 Simulação do algoritmo em Verilog HDL

Por forma a ilustrar o funcionamento e a implementação do algoritmo foram feitas duas simulações, impondo no módulo do oscilador (equação 4.16), num caso o valor de  $K_{vco}$   $2\times$  inferior, e no outro  $K_{vco}$   $1,5\times$  superior.



Tabela 4.3: Contagem do n° de ciclos de relógio em relação ao relógio de referência  $F_{ref}$ .

Iteração	$V_{ctrl_1}$	$V_{ctrl_2}$	$V_{ctrl_3}$	$V_{ctrl_4}$	$V_{ctrl_5}$
0	0	0	0	0	0
1	162	372	582	792	1002
2	144	330	518	704	890
3	130	297	466	634	801
4	117	271	423	576	729
5	99	229	339	487	617
6	112	258	403	548	694
7	104	239	374	510	644
8	115	266	416	566	715
9	108	248	388	528	668

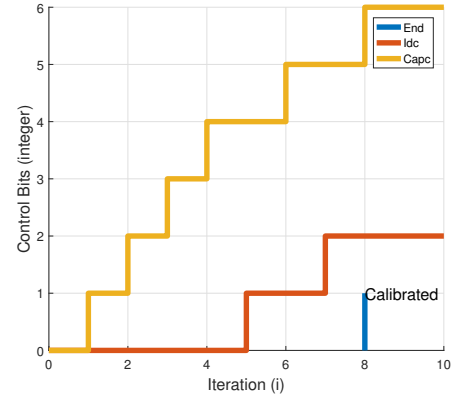


Figura 4.9: Passo de calibração para  $K_{vco}$   $1,5\times$  superior, modo=5.

É possível observar que devido ao aumento de  $Capc$  na iteração 5, é obtido um  $K_{vco}$  inferior ao pretendido para a zona  $Z_2$ , o que implica que é necessário aumentar em  $Idc$ . A combinação final apresentada na iteração 9 oferece as condições favoráveis para se operar na zona  $Z_2$ , sendo o algoritmo terminado com sucesso, ligando a *flag* associada à decisão "Calibrado" da figura 4.6.

Tabela 4.4: Contagem do N° de ciclos de relógio em relação ao relógio de referência  $F_{ref}$ .

Iteração	$V_{ctrl_1}$	$V_{ctrl_2}$	$V_{ctrl_3}$	$V_{ctrl_4}$	$V_{ctrl_5}$
0	0	0	0	0	0
1	54	124	194	264	334
2	60	140	218	297	376
3	67	155	243	330	417
4	74	170	267	363	460
5	88	201	315	429	543
6	94	217	340	462	584
7	101	232	364	495	627
8	108	248	388	528	668

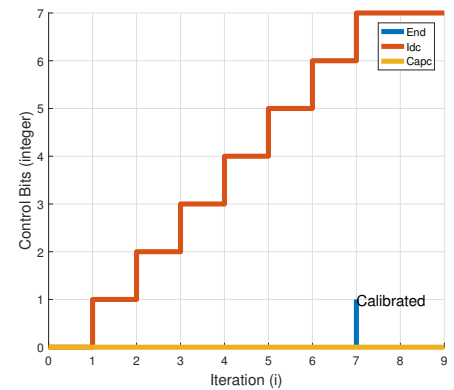


Figura 4.10: Passo de calibração para  $K_{vco}$   $2\times$  inferior, modo=0.

Neste exemplo é mostrado o caso extremo do algoritmo para o  $K_{vco}$  mínimo, sendo que dado a configuração apresentada para a comutação dos *switch* que controlam o vetor  $Idc$ , é apresentado um crescimento linear, correspondendo à ligação de todos os elementos de  $Idc$ .

## 4.2 Avaliação da cobertura da calibração

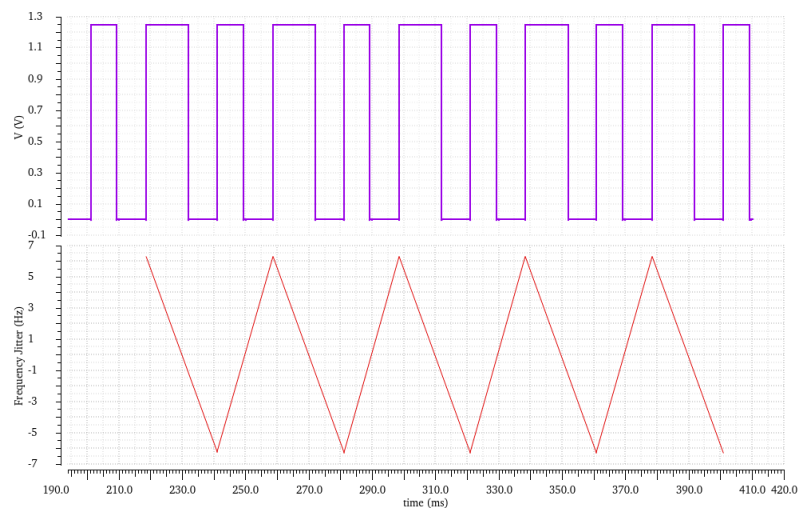
O algoritmo desenvolvido permite uma calibração *offline* do bloco do VCO, por forma a condicionar um início correto do funcionamento da PLL. As variações máximas do valor do  $K_{vco}$ , apresentadas pelas curvas definidas pelos vetores  $f_{Min}$  e  $f_{Max}$ , caracterizam os desvios superior e inferior máximos que o algoritmo consegue calibrar, sendo portanto estes os valores a integrar na qualidade da cobertura da calibração do algoritmo. É possível então cobrir, aproximadamente, 90% dos casos de um  $K_{vco}$  superior e 90% do  $K_{vco}$  inferior, sendo esta percentagem considerada relativamente aos piores casos — mais prováveis — do cenário total das variações PVT apresentadas da figura 4.4.

Para ilustrar esta calibração e a melhoria do desempenho da PLL foi feita uma comparação entre as respostas obtidas (em Spectre) antes de calibração e após calibração com a parametrização esperada após se atingir do  $K_{vco}$  da zona  $Z_2$ . A simulação apresentada corresponde a uma variação PVT, a qual é equivalente à variação de  $\times 1,5$  do  $K_{vco}$ , tomando as variações de processo para o canto FF, uma variação de temperatura para  $0^\circ\text{C}$  e um desvio da tensão de alimentação de 50 mV ( $\approx 4\%$ ) para o conjunto total da PLL. É possível observar que em regime estacionário, o desvio de frequência atinge um intervalo de variações entre 43,5 Hz e 56,5 Hz devido a variações PVT 4.11a, muito maior do que era apresentado no caso nominal da figura 3.37.

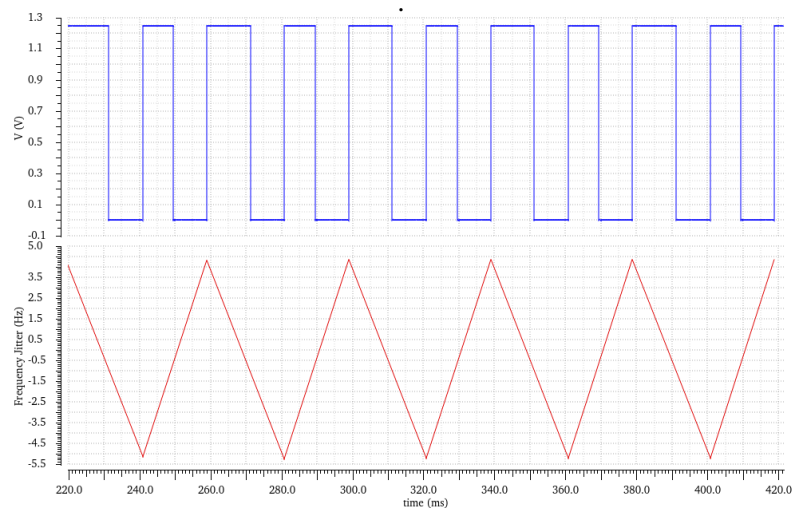
Após calibração e usando o código  $Idc=2$  e  $Capc=6$  como feito no exemplo da secção 4.1.3.4, o oscilador apresenta a curva característica da figura 4.12b. Pode ver-se que o ganho  $K_{vco}$  é melhorado, delimitado pelas zonas previamente definidas após calibração, sendo que o desvio em relação ao objetivo deve-se ao facto de que o  $K_{vco}$  obtido em 4.12a não é exatamente  $1,5\times$  superior ao do obtido na zona  $Z_2$ , como foi feito na simulação em Verilog HDL 4.9.

O código  $Idc=2$  aumenta em 40 nm o comprimento total de cada transístor NMOS e em 190 nm os PMOS; o código  $Capc=6$  implica um aumento de 7,5 pF em cada capacidade, isto para a célula de calibração proposta na figura 4.5

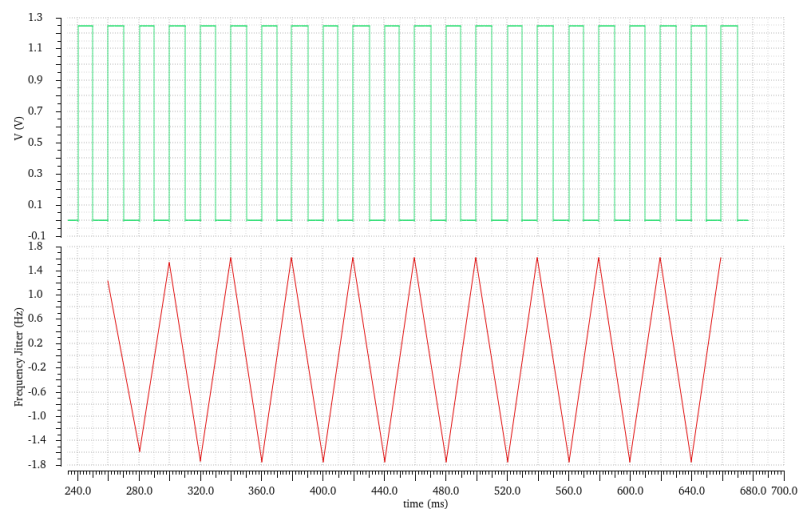
O algoritmo permite uma melhoria notória do desvio de frequência em relação à apresentada por a implementações mais complexas, e. g., como as apresentadas em [9] na secção 2.2.4. A correção apresentada em 4.11b e 4.12b conduz a um melhoramento de  $\approx 33\%$  do *jitter*, praticamente o dobro do apresentado no artigo [9] onde se apresenta um melhoramento de 17%. Com a inclusão da comutação de  $N_{div}$ , isto é, aumento da cadeia com mais um divisor caso o  $K_{vco}$  seja elevado, o *jitter* é ainda menor, apresentando um desvio máximo de  $\approx 1,5$  Hz como é visto na figura 4.13.

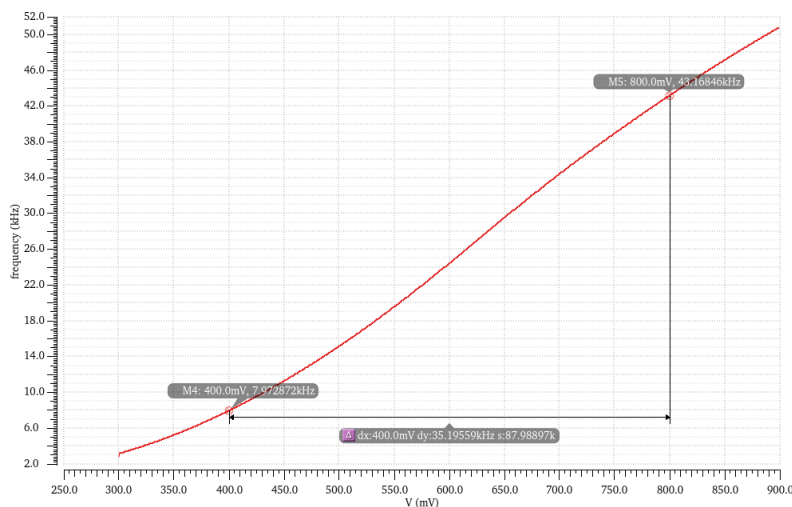


(a) Antes da calibração

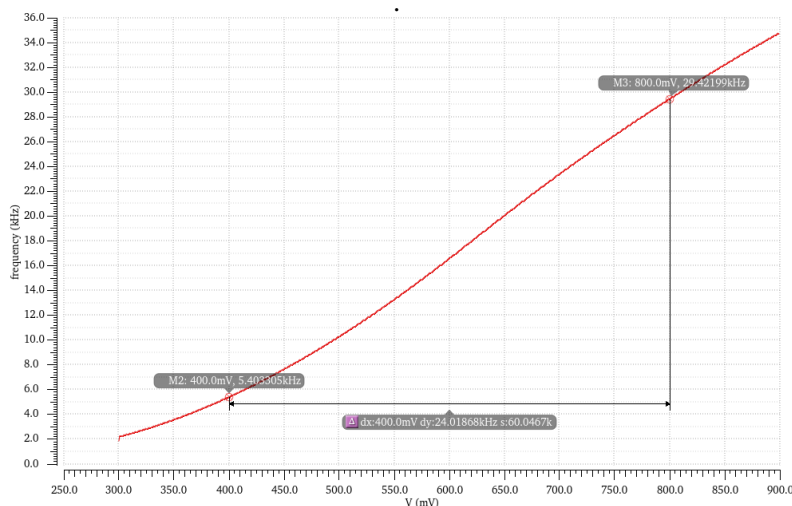


(b) Após calibração.

Figura 4.11: Melhoria do *jitter* observado à frequência de 50 Hz, devido a efeitos PVT.Figura 4.13: Melhoramento do *jitter* com a ativação de mais um divisor na cadeia e a reprogramação do VCO.



(a) Antes da calibração



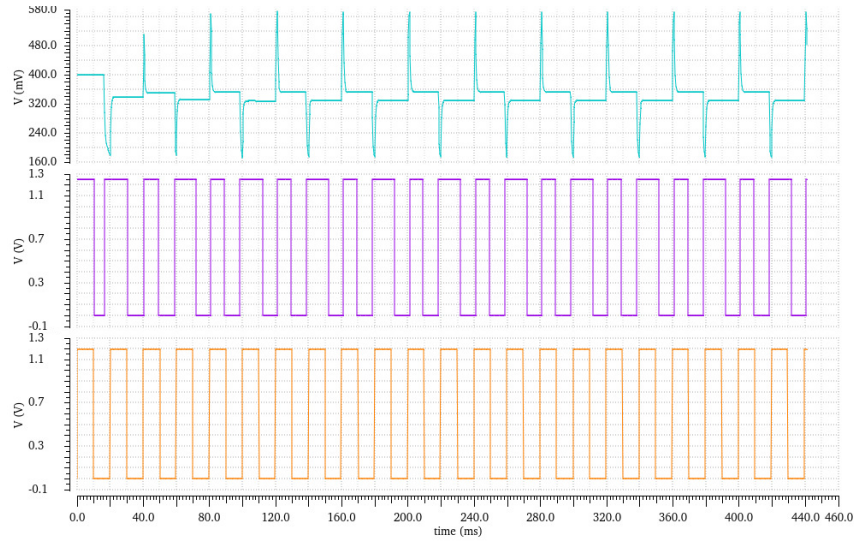
(b) Após calibração.

Figura 4.12: Calibração do ganho  $K_{VCO}$  após ajuste de  $I_{dc}$  e  $C_{apc}$ .

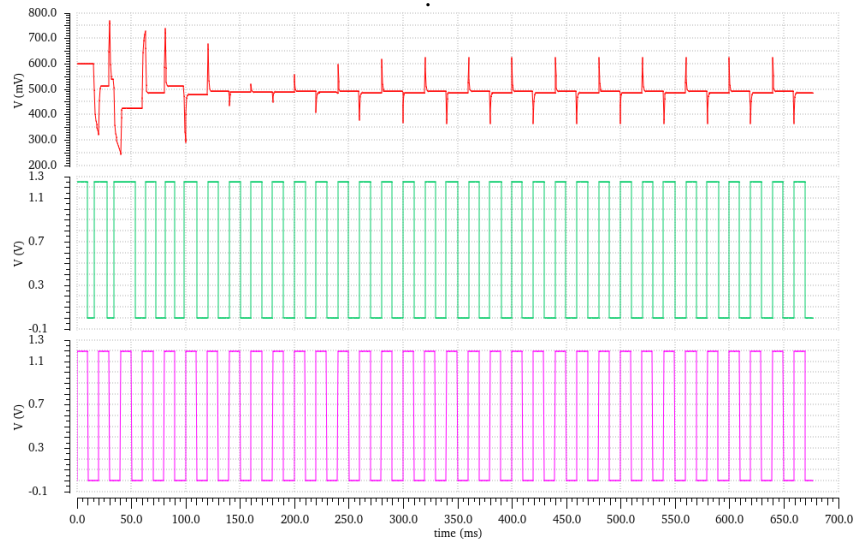
A abordagem proposta permite também uma menor área de implantação. O circuito de calibração associado à reprogramação do VCO é muito mais reduzido, comparando com os blocos VCO bias apresentados em [9] e em [11], devido a estes não considerarem no dimensionamento à previsão do comportamentos esperado para as variações do  $K_{VCO}$ , o que aqui é feito recorrendo aos algoritmos apresentados antes neste capítulo. No entanto o bloco ao qual está presente o sistema de decisão das zonas e de contagem da frequência pode apresentar uma maior desvantagem para a técnica proposta, sendo daí ser necessário ter dados concretos sobre a área associada das técnicas usadas na literatura atual para fazer uma comparação correta sem estimação.

No artigo [11] o uso do banco de capacidades, ao qual, a cada bit corresponde a comutação de uma capacidade — todas com o mesmo valor — pode ser otimizado. A seleção proposta neste trabalho permite através da combinação (como referido na matriz 4.14) de um conjunto reduzido para obter o mesmo efeito. É portanto também, melhorado o tempo de convergência do algoritmo.

A maior desvantagem do algoritmo proposto neste trabalho face ao apresentado em [9] reside no facto da calibração ser feita *offline*. Deste modo os valores de ajustes não conduzem a um melhoramento do *jitter* caso haja uma grande alteração de temperatura/tensão durante um funcionamento longo do sistema. Para se conseguir uma calibração mais eficiente de desvios PVT, seria necessário conciliar ajustes feitos *offline* e *online*.



(a) Antes da calibração



(b) Após calibração.

Figura 4.14: Melhoramento do desempenho da PLL, decorrente da calibração do  $K_{vco}$  e comutação de Ndiv.

É então mostrado na figura 4.14 a compensação feita, em que apesar de que as variações PVT no bloco da *charge-pump* forcem uma grande variação na tensão de  $V_{ctrl}$  à entrada do oscilador, este juntamente com a adição de um divisor, relaxa a PLL, nomeadamente os parâmetros dinâmicos, mais propriamente a frequência normal do sistema  $\omega_n$ , diminuindo assim o *jitter* à saída.



## Capítulo 5

# Conclusões e Trabalho Futuro

O impacto de variações de processo, temperatura e tensão em circuitos analógicos e mistos torna-se cada vez mais crítico com a utilização de tecnologias CMOS de tamanhos submicrométricos. É proposta neste trabalho uma abordagem para avaliar a sensibilidade de uma PLL a essas variações e, conseqüentemente, definir o respetivo processo de calibração. Dado que o desempenho da PLL é diretamente relacionado com a função de transferência, que exprime uma relação com parâmetros fundamentais dos respetivos blocos, é necessário avaliar como agir por forma a diminuir a sua variabilidade, sendo necessário aplicar uma seleção das melhores *features* que se podem usar para tal.

A metodologia de calibração feita neste trabalho é apresentada em dois passos, sendo inicialmente feita a recolha e processamento de um grande conjunto de dados referentes ao ganho do VCO, por forma a obter o melhor modelo que os represente com recurso ao treino de um modelo baseado em um processo de regressão Gaussiano. Recorrendo aos resultados do modelo, é criado um algoritmo de calibração baseado na definição de zonas delimitadas do funcionamento do VCO, às quais é atribuído um conjunto de valores da corrente e capacidade do VCO que permitem ajustar a operação. Para o efeito foram desenhados um VCO programável, e um bloco lógico que constitui o controlador que implementa o algoritmo referido.

Pode-se concluir que nem sempre um sobredimensionamento (*overdesign*) de circuitos analógicos-mistos é o ideal para criar o sistema imune a efeitos PVTs, dado que leva por vezes à obtenção de circuitos com um consumo/área superior àquele que se pode obter recorrendo a circuitos auxiliares de calibração por forma a avaliar e corrigir as condições de funcionamento do sistema em causa.

O uso e aplicação de métodos probabilísticos, algoritmos baseados em aprendizagem automática e reconhecimento padrões, por forma a caracterizar e prever quais os pontos de calibração, é portanto, uma grande valia para o melhoramento de sistemas de autoteste e de autocalibração.

## 5.1 Trabalho Futuro

Os blocos da PLL apresentados neste trabalho foram desenhados e validados por simulação. Para se conseguir o seu fabrico faltam ainda os últimos passos do projeto, sendo eles o (*layout*) e simulações post-layout, por forma a avaliar o efeito de capacidades parasitas nos blocos e outras linearidades resultantes da interligação dos elementos.

Apesar da baixa frequência de funcionamento e a existência de um filtro passa-baixo, com banda de passagem muito reduzida, uma avaliação do ruído à frequência de *offset* do oscilador é também importante, por forma a analisar o impacto deste [57].

Quanto ao método de calibração, é necessário adicionar uma maior resolução de dados para efetivamente aumentar a sua resolução. Isto implica fazer um estudo das variações PVT para o bloco da *charge-pump*, assim como encontrar uma forma de o calibrar, operação mais difícil dado que os circuitos para leitura de corrente são geralmente mais complexos.

Por forma a melhorar também o método de calibração, nomeadamente via uma abordagem *on-line*, esta é possível aproveitando os recursos e dados obtidos. Para tal seria necessário um estudo e avaliação do comportamento do VCO integrado na PLL, por forma a caracterizar os desvios máximos do *jitter* em frequência.



## Anexo A

# Resultados implementação com ganho do oscilador inferior ao da implementação final

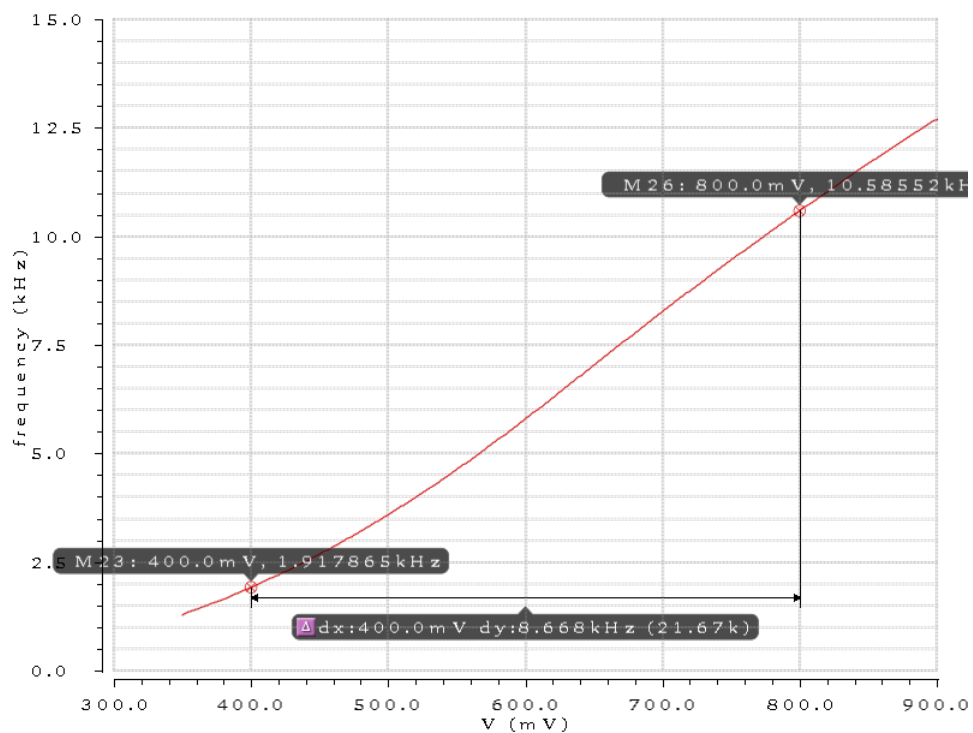


Figura A.1:  $K_{vco} = 21.67 \text{ KHz/V}$ .

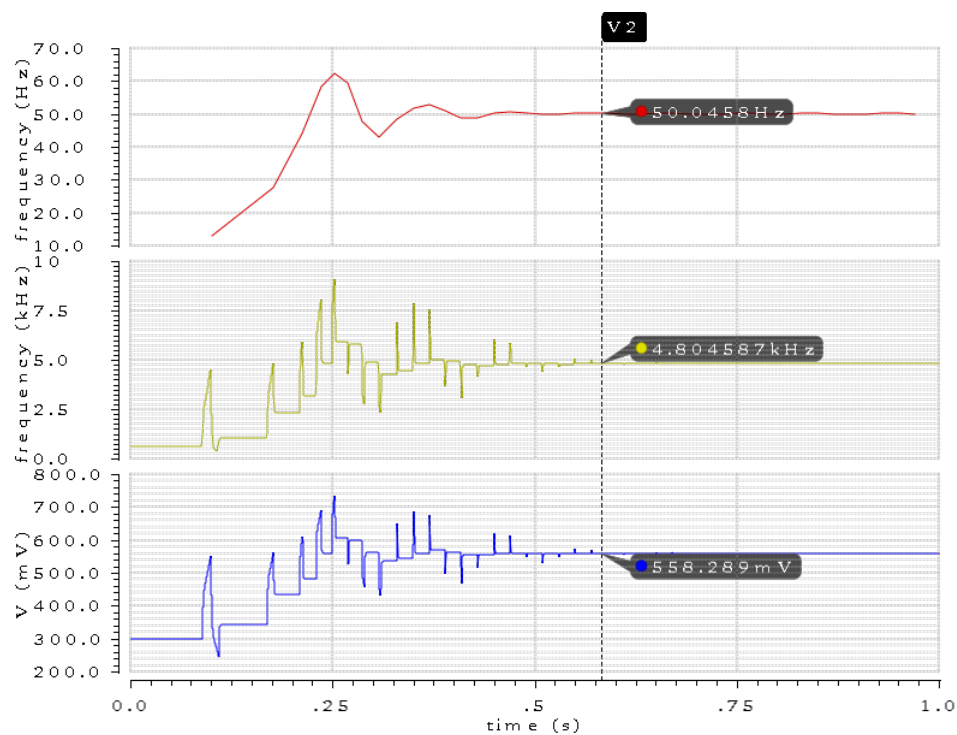
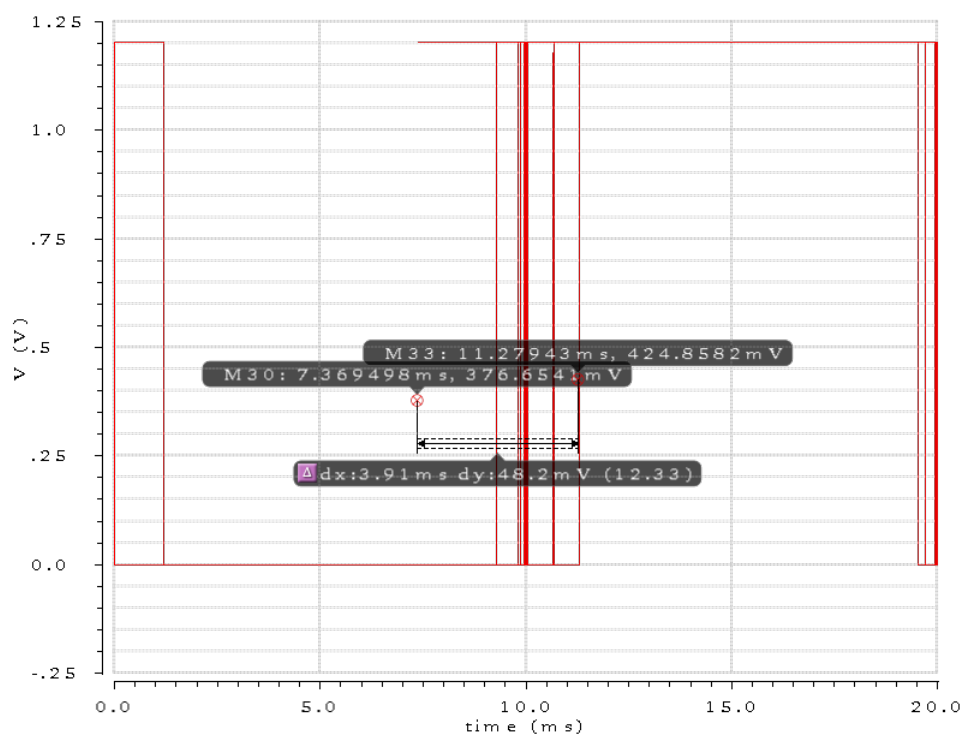


Figura A.2: Resposta ao degrau.

Figura A.3: Incerteza do instante de ocorrência da transição para o mesmo tempo de assentamento do  $K_{vco} = 65\text{KHz/V}$  à frequência de 50 Hz.

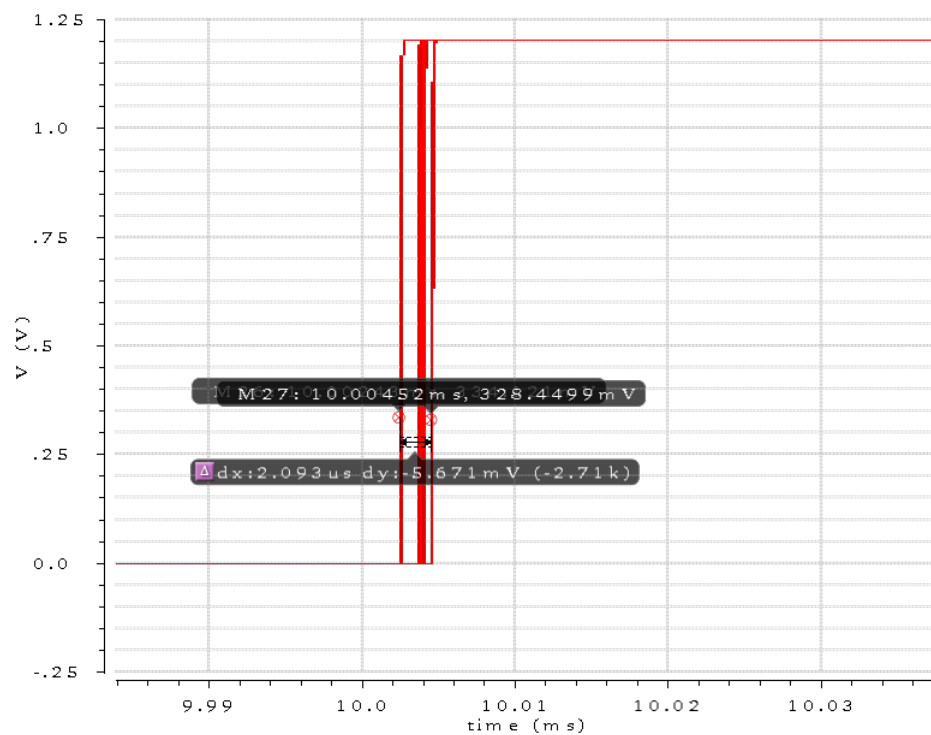


Figura A.4: Incerteza do instante de ocorrência da transição em *steady-state* à frequência de 50 Hz .

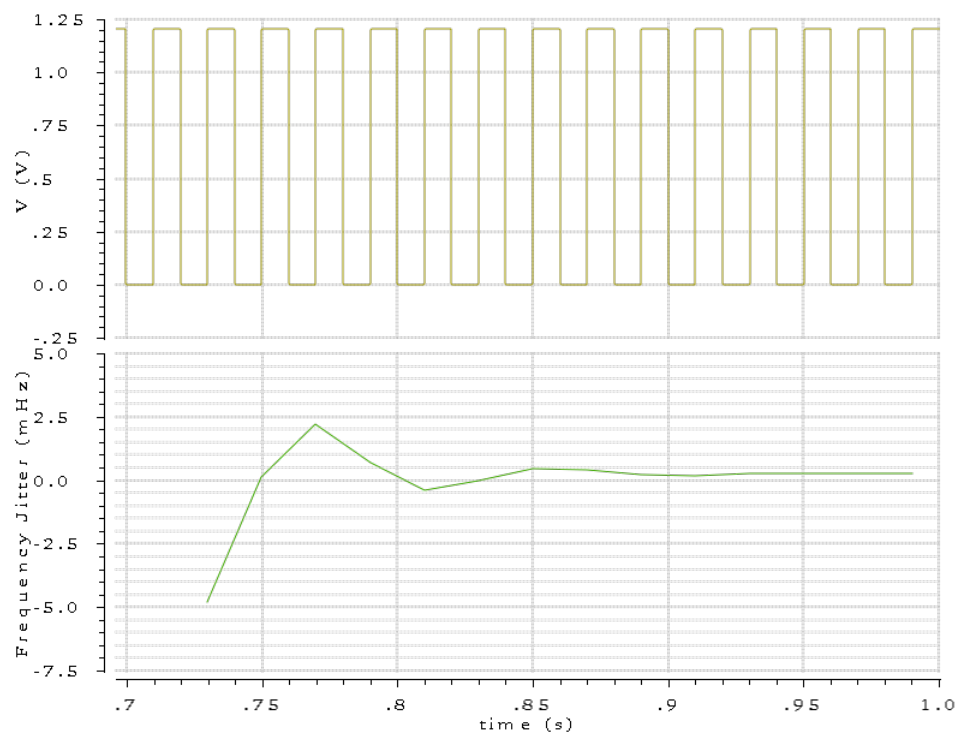


Figura A.5: *Jitter* de frequência em *steady-state* .

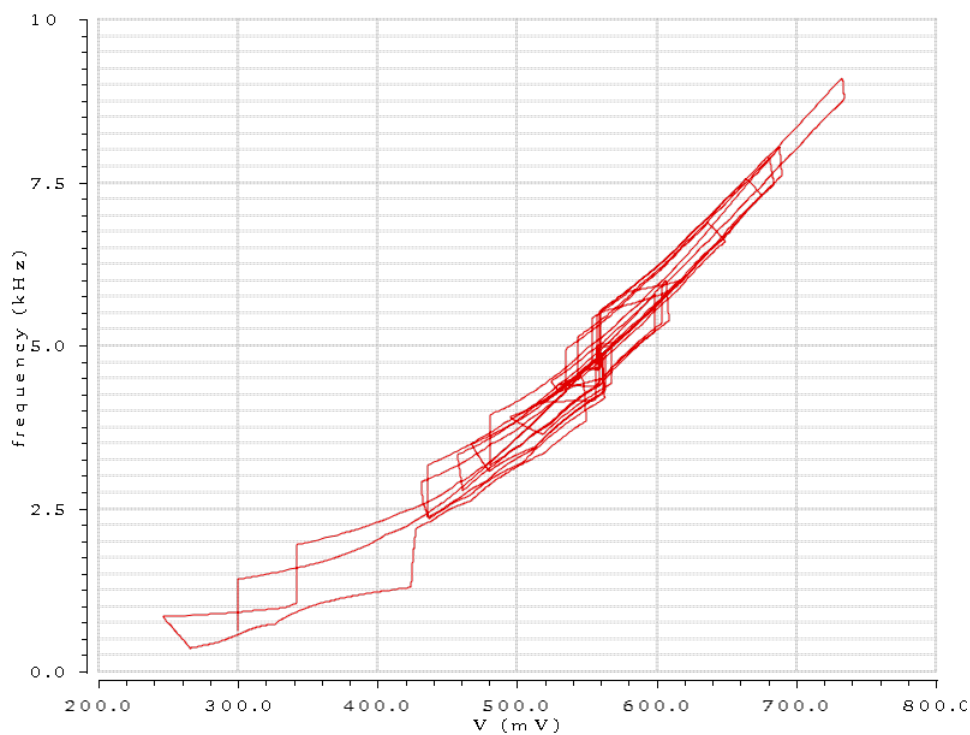


Figura A.6: Variação da frequência no oscilador durante a resposta transitória.

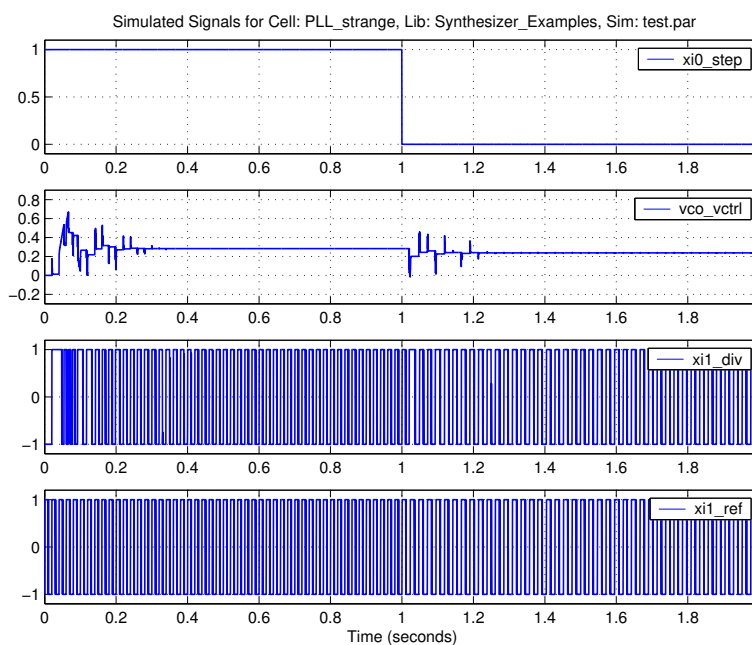


Figura A.7: Banda máxima do sinal à entrada, com o exemplo de sinal com degrau de 9Hz para a implementação com maior banda da PLL.

## Anexo B

# Resultados simulação Verilog HDL

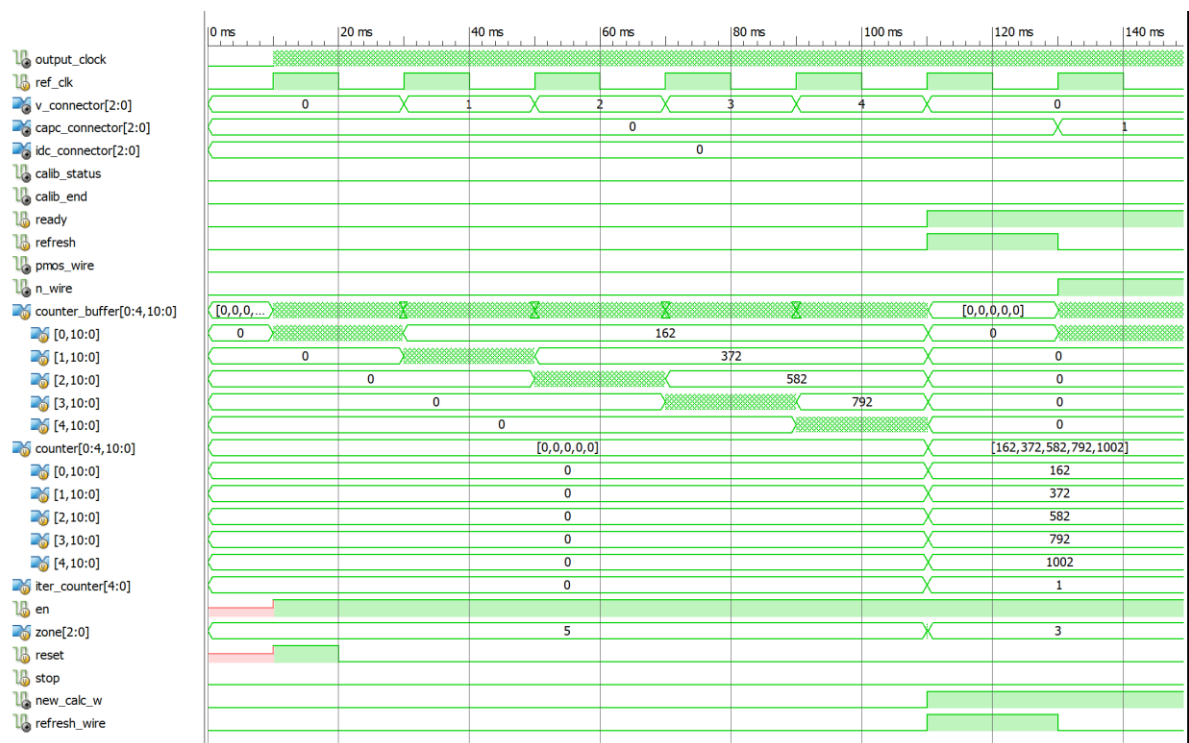
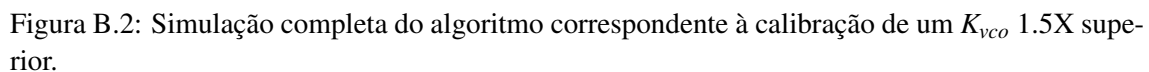


Figura B.1: Primeira iteração do algoritmo correspondente à calibração de um  $K_{vco}$  1.5X superior



# Bibliografia

- [1] Texas Instruments Incorporated. An-1261 power supply effects on noise performance. Technical report, Outubro 2000. Application Report, Revisto a Maio de 2015.
- [2] Paul Ampadu David Wolpert. *Managing Temperature Effects in Nanoscale Adaptive Systems*. Springer-Verlag New York, First edition, 2012.
- [3] Sleiman Bou Sleiman. Built-in-self-test and digital self-calibration for radio frequency. Master's thesis, The Ohio State University, USA, 2011. Dissertation for Graduate Program in Electrical and Computer Engineering.
- [4] Martin Andraud. *Solutions for the self-adaptation of wireless systems*. PhD thesis, Micro et nanotechnologies/Microélectronique. Université Grenoble Alpes, 2016.
- [5] Abbas Ramazani, Sadegh Biabani, and Gholamreza Hadidi. Cmos ring oscillator with combined delay stages. *AEU - International Journal of Electronics and Communications*, 68(6):515 – 519, 2014.
- [6] Yin Xizhen, Xiao Shimao, Jin Yuhua, Wu Qiwu, Ma Chengyan, and Ye Tianchun. A constant loop bandwidth fractional-n frequency synthesizer for gnss receivers. 33, 04 2012.
- [7] C. Stroud. elec6970 overview of bist. elec6970@Auburn University , disponível em <http://www.eng.auburn.edu/~strouce/class/elec6970/BISTc1.pdf>, acessado a última vez em 17 de Junho de 2018.
- [8] Kuncup Iswandy and Andreas Koenig. Pso for fault-tolerant nearest neighbor classification employing reconfigurable, analog hardware implementation in low power intelligent sensor systems. Technical report, Institute of Integrated Sensor Systems, University of Kaiserslautern, Germany, 2008. Eighth International Conference on Hybrid Intelligent Systems.
- [9] K. T. Khachikyan, A. H. Balabanyan, and A. S. Petrosyan. Pll control voltage stabilization method for high-speed systems. In *2016 XXV International Scientific Conference Electronics (ET)*, pages 1–4, sep 2016.
- [10] D. M. Fischette, A. L. S. Loke, R. J. DeSantis, and G. R. Talbot. An embedded all-digital circuit to measure pll response. *IEEE Journal of Solid-State Circuits*, 45(8):1492–1503, Agosto 2010.

- [11] Jaewook Shin and Hyunchol Shin. Design considerations for autocalibrations of wide-band  $\Delta\Sigma$  fractional-n pll synthesizers. *JECE*, 2011:1:1–1:9, Janeiro 2011.
- [12] J. Lee, S. Bhagavatula, K. Roy, and B. Jung. Self-healing design in deep scaled cmos technologies. In *2011 IEEE 54th International Midwest Symposium on Circuits and Systems (MWSCAS)*, pages 1–4, Agosto 2011.
- [13] Ricardo Gutierrez-Osuna. Csce 666 pattern analysis. CSE@TAMU , disponível em [http://research.cs.tamu.edu/prism/lectures/pr/pr\\_l11.pdf](http://research.cs.tamu.edu/prism/lectures/pr/pr_l11.pdf), acedido a última vez em 17 de Junho de 2018.
- [14] Hyung Kyu Lee and Shingo Yoneoka. A machine learning approach to the frequency control of mems resonators, cs229 autumn 2008. Stanford University, California, USA, disponível em <http://citeseer.ist.psu.edu/viewdoc/download?doi=10.1.1.217.3142&rep=rep1&type=pdf>, acedido a última vez em 17 de Junho de 2018.
- [15] M.H. Perrott. Phase-locked loops and their applications. Cppsim@MIT , disponível em [http://www.cppsim.com/pll\\_lectures.html](http://www.cppsim.com/pll_lectures.html), acedido a última vez em 17 de Junho de 2018.
- [16] A. Koithyar and T. K. Ramesh. Integer-n charge pump phase locked loop with reduced current mismatch. In *2017 International Conference on Wireless Communications, Signal Processing and Networking (WiSPNET)*, pages 650–653, Março 2017.
- [17] F. Gardner. Charge-pump phase-lock loops. *IEEE Transactions on Communications*, 28(11):1849–1858, Novembro 1980.
- [18] H. Babazadeh, A. Esmaili, and K. Hadidi. A high-speed and wide detectable frequency range phase detector for dlls. In *2009 NORCHIP*, pages 1–3, Novembro 2009.
- [19] Jung-Woong Park, Ho-Yong Choi, and Nam-Soo Kim. Two-stage feedback-looped charge-pump for spur reduction in cmos pll. *Analog Integr. Circuits Signal Process.*, 83(2):143–148, May 2015.
- [20] Cleon Petty and Paul. Shockman. Application note : Odd number divide by counters with 50% outputs and synchronous clocks, 1999.
- [21] R. Jacob Baker. *CMOS Circuit Design, Layout, and Simulation*. Wiley-IEEE Press, 3rd edition, 2010.
- [22] W. Shockley. Problems related to p-n junctions in silicon. *Czechoslovak Journal of Physics*, 11(2):81–121, Fevereiro 1961.
- [23] Neil Weste and David Harris. *CMOS VLSI Design: A Circuits and Systems Perspective*. Addison-Wesley Publishing Company, USA, 4th edition, 2010.



- [24] D. Banerjee. *PlI Performance, Simulation, and Design 5th Edition*. DOG EAR PUB LLC, 2017.
- [25] Christian Munker. *Spectral PLL Built-In Self-Test for Integrated Cellular Transmitters*. PhD thesis, Der technischen Fakultät der Universität Erlangen-Nürnberg, Março 2010.
- [26] N. S. Altman. An introduction to kernel and nearest-neighbor nonparametric regression. *The American Statistician*, 46(3):175–185, 1992.
- [27] Rabi Behera and Kajaree Das. A survey on machine learning: Concept, algorithms and applications. 2, Fevereiro 2017.
- [28] Sonika Jindal Sunpreet Kaur. A survey on machine learning algorithms. *International Journal of Innovative Research in Advanced Engineering*, Issue 11, Volume 3, Novembro 2006.
- [29] S. B. Kotsiantis. Supervised machine learning: A review of classification techniques. In *Proceedings of the 2007 Conference on Emerging Artificial Intelligence Applications in Computer Engineering: Real Word AI Systems with Applications in eHealth, HCI, Information Retrieval and Pervasive Technologies*, pages 3–24, Amsterdam, The Netherlands, The Netherlands, 2007. IOS Press.
- [30] Isabelle Guyon and André Elisseeff. An introduction to variable and feature selection. *J. Mach. Learn. Res.*, 3:1157–1182, 2003.
- [31] Andrew Gordon Wilson and Ryan Prescott Adams. Gaussian process kernels for pattern discovery and extrapolation. In *Proceedings of the 30th International Conference on International Conference on Machine Learning - Volume 28, ICML'13*, pages III–1067–III–1075. JMLR.org, 2013.
- [32] Christopher M. Bishop. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer-Verlag, Berlin, Heidelberg, 2006.
- [33] Richard Schwartz Steven Austin and Paul Placeways. The forward-backward search algorithm. Technical report, BBN Sytems and Technologies Cambridge MA 02138, 1991.
- [34] J. Kittler P. Pudil \*, J. Novovicova. Floating search methods in feature selection. *Pattern Recognition Letters* 15, pages 1119–1125, Novembro 1994.
- [35] Girish Chandrashekar and Ferat Sahin. A survey on feature selection methods. *Computers and Electrical Engineering*, 40(1):16 – 28, 2014. 40th-year commemorative issue.
- [36] Razieh Sheikhpour, Mehdi Agha Sarram, Sajjad Gharaghani, and Mohammad Ali Zare Chahooki. A survey on semi-supervised feature selection methods. *Pattern Recognition*, 64:141 – 158, 2017.
- [37] C. T. C. Nguyen. Mems technology for timing and frequency control. *IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control*, 54(2):251–270, Fevereiro 2007.

- [38] J. A. Tierno, A. V. Rylyakov, and D. J. Friedman. A wide power supply range, wide tuning range, all static cmos all digital pll in 65 nm soi. *IEEE Journal of Solid-State Circuits*, 43(1):42–51, Janeiro 2008.
- [39] K Sithamparanathan. A software phase locked loop from theory to practice: Tms320c6000 dsp based implementation and analysis. In Dr Johnson Agbinya, editor, *Proceedings of AusWireless conference 2006*, pages 1–6. University of Technology, Sydney, Março 2006.
- [40] Mozhgan Mansuri and IEEE Chih-Kong Ken Yang, Member. Jitter optimization based on phase-locked loop design parameters. *IEEE JOURNAL OF SOLID-STATE CIRCUITS, VOL. 37, NO. 11*, pages 1375–1382, Novembro 2002.
- [41] Prof. Ali M. Niknejad. Phase locked loops (pll) and frequency synthesis. U.C. Berkeley, Los Angeles, disponível em [rfic.eecs.berkeley.edu/ee242/pdf/Module\\_7\\_3\\_PLL.pdf](http://rfic.eecs.berkeley.edu/ee242/pdf/Module_7_3_PLL.pdf), acessado a última vez em 17 de Junho de 2018.
- [42] M. Kozak and E. G. Friedman. Design and simulation of fractional-n pll frequency synthesizers. In *2004 IEEE International Symposium on Circuits and Systems (IEEE Cat. No.04CH37512)*, volume 4, pages IV–780–3 Vol.4, Maio 2004.
- [43] Best R.E. *Phase-Locked Loops Design, Simulation, and Applications*. McGraw-Hill, Fifth edition, 2003.
- [44] G. Bhargav, G. Prasad, S. Datta Canchi, and B. Chanikya. Design and analysis of phase locked loop in 90nm cmos. In *2016 Thirteenth International Conference on Wireless and Optical Communications Networks (WOCN)*, pages 1–7, Julho 2016.
- [45] Sung-Mo (Steve) Kang and Yusuf Leblebici. *CMOS Digital Integrated Circuits Analysis & Design*. McGraw-Hill, Inc., New York, NY, USA, 3 edition, 2003.
- [46] Allen Phillip E. Allen, Douglas R. Holberg. *CMOS Analog Circuit Design*. Oxford University press, Second edition, 2002.
- [47] X. Li, J. Zheng, and Li Zhao. Sigma-delta fractional-n frequency synthesis. In *2006 International Conference on Communication Technology*, pages 1–2, Novembro 2006.
- [48] Kruti P. Thakore Kashyap K. Patel, Nilesh D. Patel. Charge pump, loop filter and vco for phase lock loop using 0.18um cmos technology. *IOSR Journal of VLSI and Signal Processing (IOSR-JVSP)*, pages 21–25, Maio-Junho 2013.
- [49] Mike Curtin. Design a direct 6-ghz local oscillator with a wideband integer-n pll synthesizer. Technical report, Analog Dialogue 35-6, 2001.
- [50] D.Anitha K.A.Jyotsna. Design of high performance pll using process,temperature compensated vco. *International Conference on VLSI, Communication and Instrumentation*, pages 15–20, 2011.

- [51] J. R. Loo-Yau L. Guerrero-Linares, F. Sandoval-Ibarra. Non-idealities in analog circuits design: What does it really mean? 978-1-4673-2527-1/12 *IEEE*, pages 586–589, Maio-Junho 2012.
- [52] P. L. Ramos, J. M. da Silva, D. R. Ferreira, and M. B. Santos. Statistically enhanced analogue and mixed-signal design and test. In *2016 IEEE 21st International Mixed-Signal Testing Workshop (IMSTW)*, pages 1–5, Julho 2016.
- [53] John Duchi Andrew Ng. Cs229 machine learning, 2016. Stanford University, California, USA, disponível em <http://cs229.stanford.edu/>, acedido a última vez em 17 de Junho de 2018.
- [54] Carl Edward Rasmussen and Christopher K. I. Williams. *Gaussian processes for machine learning*. Adaptive computation and machine learning. MIT Press, 2006.
- [55] V. Melikyan, A. Balabanyan, A. Durgaryan, H. Stepanyan, K. Sloyan, H. Musayelyan, and G. Markosyan. Pvt variation detection and compensation methods for high-speed systems. In *2013 IFIP/IEEE 21st International Conference on Very Large Scale Integration (VLSI-SoC)*, pages 322–327, Outubro 2013.
- [56] D.J. Smith. *HDL Chip Design: A Practical Guide for Designing, Synthesizing, and Simulating ASICs and FPGAs Using VHDL Or Verilog*. Doone Publications, 1996.
- [57] IBM. Model pll dynamics and phase-noise performance, 1999. University of California, Los Angeles , disponível em [www.ee.ucla.edu/~brweb/teaching/215C\\_W2013/PLLs.pdf](http://www.ee.ucla.edu/~brweb/teaching/215C_W2013/PLLs.pdf), acedido a última vez em 17 de Junho de 2018.